

# A PTAS for a Class of Stochastic Dynamic Programs \*

Hao Fu <sup>†1</sup>, Jian Li <sup>‡1</sup>, and Pan Xu <sup>§2</sup>

<sup>1</sup> Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China.

<sup>2</sup> Department of Computer Science, University of Maryland, College Park, USA.

May 22, 2018

## Abstract

We develop a framework for obtaining polynomial time approximation schemes (PTAS) for a class of stochastic dynamic programs. Using our framework, we obtain the first PTAS for the following stochastic combinatorial optimization problems:

1. *Probemax* [20]: We are given a set of  $n$  items, each item  $i \in [n]$  has a value  $X_i$  which is an independent random variable with a known (discrete) distribution  $\pi_i$ . We can *probe* a subset  $P \subseteq [n]$  of items sequentially. Each time after probing an item  $i$ , we observe its value realization, which follows the distribution  $\pi_i$ . We can *adaptively* probe at most  $m$  items and each item can be probed at most once. The reward is the maximum among the  $m$  realized values. Our goal is to design an adaptive probing policy such that the expected value of the reward is maximized. To the best of our knowledge, the best known approximation ratio is  $1 - 1/e$ , due to Asadpour *et al.* [2]. We also obtain PTAS for some generalizations and variants of the problem.
2. *Committed Pandora's Box* [25, 23]: We are given a set of  $n$  boxes. For each box  $i \in [n]$ , the cost  $c_i$  is deterministic and the value  $X_i$  is an independent random variable with a known (discrete) distribution  $\pi_i$ . Opening a box  $i$  incurs a cost of  $c_i$ . We can adaptively choose to open the boxes (and observe their values) or stop. We want to maximize the expectation of the realized value of the last opened box minus the total opening cost.
3. *Stochastic Target* [16]: Given a predetermined target  $\mathbb{T}$  and  $n$  items, we can adaptively insert the items into a knapsack and insert at most  $m$  items. Each item  $i$  has a value  $X_i$  which is an independent random variable with a known (discrete) distribution. Our goal is to design an adaptive policy such that the probability of the total values of all items inserted being larger than or equal to  $\mathbb{T}$  is maximized. We provide the first bi-criteria PTAS for the problem.
4. *Stochastic Blackjack Knapsack* [17]: We are given a knapsack of capacity  $\mathbb{C}$  and probability distributions of  $n$  independent random variables  $X_i$ . Each item  $i \in [n]$  has a size  $X_i$  and a profit  $p_i$ . We can adaptively insert the items into a knapsack, as long as the capacity constraint is not violated. We want to maximize the expected total profit of all inserted items. If the capacity constraint is violated, we lose all the profit. We provide the first bi-criteria PTAS for the problem.

---

\*This work is published in the 45th International Colloquium on Automata, Languages, and Programming (ICALP2018) [10]. This research is supported in part by the National Basic Research Program of China Grant 2015CB358700, the National Natural Science Foundation of China Grant 61772297, 61632016, 61761146003, and a grant from Microsoft Research Asia.

<sup>†</sup>Email: fu-h13@mails.tsinghua.edu.cn

<sup>‡</sup>Email: lijian83@mail.tsinghua.edu.cn

<sup>§</sup>Email: panxu@cs.umd.edu

# 1 Introduction

Consider an online stochastic optimization problem with a finite number of rounds. There are a set of tasks (or items, boxes, jobs or actions). In each round, we can choose a task and each task can be chosen at most once. We have an initial “state” of the system (called the value of the system). At each time period, we can select a task. Finishing the task generates some (possibly stochastic) feedback, including changing the value of the system and providing some profit for the round. Our goal is to design a strategy to maximize our total (expected) profit.

The above problem can be modeled as a class of stochastic dynamic programs which was introduced by Bellman [3]. There are many problems in stochastic combinatorial optimization which fit in this model, *e.g.*, the stochastic knapsack problem [9], the Probemax problem [20]. Formally, the problem is specified by a 5-tuple  $(\mathcal{V}, \mathcal{A}, f, g, h, T)$ . Here,  $\mathcal{V}$  is the set of all possible values of the system.  $\mathcal{A}$  is a finite set of items or tasks which can be selected and each item can be chosen at most once. This model proceeds for at most  $T$  rounds. At each round  $t \in [T]$ , we use  $I_t \in \mathcal{V}$  to denote the current value of the system and  $\mathcal{A}_t \subseteq \mathcal{A}$  the set of remaining available items. If we select an item  $a_t \in \mathcal{A}_t$ , the value of the system changes to  $f(I_t, a_t)$ . Here  $f$  may be stochastic and is assumed to be independent for each item  $a_t \in \mathcal{A}$ . Using the terminology from Markov decision processes, the state at time  $t$  is  $s_t = (I_t, \mathcal{A}_t) \in \mathcal{V} \times 2^{\mathcal{A}}$ .<sup>1</sup> Hence, if we select an item  $a_t \in \mathcal{A}_t$ , the evolution of the state is determined by the state transition function  $f$ :

$$s_{t+1} = (I_{t+1}, \mathcal{A}_{t+1}) = (f(I_t, a_t), \mathcal{A}_t \setminus a_t) \quad t = 1, \dots, T. \quad (1.1)$$

Meanwhile the system yields a random profit  $g(I_t, a_t)$ . The function  $h(I_{T+1})$  is the terminal profit function at the end of the process.

We begin with the initial state  $s_1 = (I_1, \mathcal{A})$ . We choose an item  $a_1 \in \mathcal{A}$ . Then the system yields a profit  $g(I_1, a_1)$ , and moves to the next state  $s_2 = (I_2, \mathcal{A}_2)$  where  $I_2$  follows the distribution  $f(I_1, a_1)$  and  $\mathcal{A}_2 = \mathcal{A} \setminus a_1$ . This process is iterated yielding a random sequence

$$s_1, a_1, s_2, a_2, s_3, \dots, a_T, s_{T+1}.$$

The profits are accumulated over  $T$  steps.<sup>2</sup> The goal is to find a policy that maximizes the expectation of the total profits  $\mathbb{E}\left[\sum_{t=1}^T g(I_t, a_t) + h(I_{T+1})\right]$ . Formally, we want to determine:

$$\begin{aligned} \text{DP}^*(s_1) &= \max_{\{a_1, \dots, a_T\} \subseteq \mathcal{A}} \mathbb{E}\left[\sum_{t=1}^T g(I_t, a_t) + h(I_{T+1})\right] & (\text{DP}) \\ \text{subject to: } & I_{t+1} = f(I_t, a_t), \quad t = 1, \dots, T. \end{aligned}$$

By Bellman’s equation [3], for every initial state  $s_1 = (I_1, \mathcal{A})$ , the optimal value  $\text{DP}^*(s_1)$  is given by  $\text{DP}_1(I_1, \mathcal{A})$ . Here  $\text{DP}_1$  is the function defined by  $\text{DP}_{T+1}(I_{T+1}) = h(I_{T+1})$  together with the recursion:

$$\text{DP}_t(I_t, \mathcal{A}_t) = \max_{a_t \in \mathcal{A}_t} \mathbb{E}\left[\text{DP}_{t+1}(f(I_t, a_t), \mathcal{A}_t \setminus a_t) + g(I_t, a_t)\right], \quad t = 1, \dots, T. \quad (1.2)$$

When the value and the item spaces are finite, and the expectations can be computed, this recursion yields an algorithm to compute the optimal value. However, since the state space  $\mathcal{S} = \mathcal{V} \times 2^{\mathcal{A}}$  is exponentially large, this exact algorithm requires exponential time. Since this model can capture several stochastic optimization problems which are known (or believed) to be #P-hard or even PSPACE-hard, we are interested in obtaining polynomial-time approximation algorithms with provable performance guarantees.

<sup>1</sup>This is why we do not call  $I_t$  the state of the system.

<sup>2</sup>If less than  $T$  steps, we can use some special items to fill which satisfy that  $f(I, a) = I$  and  $g(I, a) = 0$  for any value  $I \in \mathcal{V}$ .

## 1.1 Our Results

In order to obtain a polynomial time approximation scheme (PTAS) for the stochastic dynamic program, we need the following assumptions.

**Assumption 1.** *In this paper, we make the following assumptions.*

1. *The value space  $\mathcal{V}$  is discrete and ordered, and its size  $|\mathcal{V}|$  is a constant. W.l.o.g., we assume  $\mathcal{V} = (0, 1, \dots, |\mathcal{V}| - 1)$ .*
2. *The function  $f$  satisfies that  $f(I_t, a_t) \geq I_t$ , which means the value is nondecreasing.*
3. *The function  $h : \mathcal{V} \rightarrow \mathbb{R}^{\geq 0}$  is a nonnegative function. The expected profit  $\mathbb{E}[g(I_t, a_t)]$  is nonnegative (although the function  $g(I_t, a_t)$  may be negative with nonzero probability).*

Assumption (1) seems to be quite restrictive. However, for several concrete problems where the value space is not of constant size (*e.g.*, Probemax in Section 1.2), we can discretize the value space and reduce its size to a constant, without losing much profit. Assumption (2) and (3) are quite natural for many problems. Now, we state our main result.

**Theorem 1.1.** *For any fixed  $\varepsilon > 0$ , if Assumption 1 holds, we can find an adaptive policy in polynomial time  $n^{2^{O(\varepsilon^{-3})}}$  with expected profit at least  $\text{OPT} - O(\varepsilon) \cdot \text{MAX}$  where  $\text{MAX} = \max_{I \in \mathcal{V}} \text{DP}_1(I, \mathcal{A})$  and  $\text{OPT}$  denotes the expected profit of the optimal adaptive policy.*

**Our Approach:** For the stochastic dynamic program, an optimal adaptive policy  $\sigma$  can be represented as a decision tree  $\mathcal{T}$  (see Section 2 for more details). The decision tree corresponding to the optimal policy may be exponentially large and arbitrarily complicated. Hence, it is unlikely that one can even represent an optimal decision for the stochastic dynamic program in polynomial space. In order to reduce the space, we focus a special class of policies, called *block adaptive policy*. The idea of *block adaptive policy* was first introduced by Bhalgat *et al.* [6] and further generalized in [18] to the context of the stochastic knapsack. To the best of our knowledge, the idea has not been extended to other applications. In this paper, we make use of the notion of block adaptive policy as well, but we target at the development of a general framework. For this sake we provide a general model of block policy (see Section 3). Since we need to work with the more abstract dynamic program, our construction of block adaptive policy is somewhat different from that in [6, 18].

Roughly speaking, in a block adaptive policy, we take a batch of items simultaneously instead of a single one each time. This can significantly reduce the size of the decision tree. Moreover, we show that there exists a block-adaptive policy that approximates the optimal adaptive policy and has only a constant number of blocks on the decision tree (the constant depends on  $\varepsilon$ ). Since the decision tree corresponding to a block adaptive policy has a constant number of nodes, the number of all topologies of the block decision tree is a constant. Fixing the topology of the decision tree corresponding to the block adaptive policy, we still need to decide the subset of items to place in each block. Again, there is exponential number of possible choices. For each block, we can define a *signature* for it, which allows us to represent a block using polynomially many possible signatures. The signatures are so defined such that two subsets with the same signature have approximately the same reward distribution. Finally, we show that we can enumerate the signatures of all blocks in polynomial time using dynamic programming and find a nearly optimal block-adaptive policy. The high level idea is somewhat similar to that in [18], but the details are again quite different.

## 1.2 Applications

Our framework can be used to obtain the first PTAS for the following problems.

### 1.2.1 The Probemax Problem

In the Probemax problem, we are given a set of  $n$  items. Each item  $i \in [n]$  has a value  $X_i$  which is an independent random variable following a known (discrete) distribution  $\pi_i$ . We can *probe* a subset  $P \subseteq [n]$  of items sequentially. Each time after *probing* an item  $i$ , we observe its value realization, which is an independent sample from the distribution  $\pi_i$ . We can *adaptively* probe at most  $m$  items and each item can be probed at most once. The reward is the maximum among the  $m$  realized values. Our goal is to design an adaptive probing policy such that the expected value of the reward is maximized.

Despite being a very basic stochastic optimization problem, we still do not have a complete understanding of the approximability of the Probemax problem. It is not even known whether it is intractable to obtain the optimal policy. For the non-adaptive Probemax problem (*i.e.*, the probed set  $P$  is just a priori fixed set), it is easy to obtain a  $1 - 1/e$  approximation by noticing that  $f(P) = \mathbb{E}[\max_{i \in P} X_i]$  is a submodular function (see e.g., Chen *et al.* [8]). Chen *et al.* [8] obtained the first PTAS. When considering the adaptive policies, Munagala [20] provided a  $\frac{1}{8}$ -approximation ratio algorithm by LP relaxation. His policy is essentially a non-adaptive policy (it is related to the contention resolution schemes [24, 11]). They also showed that the *adaptivity gap* (the gap between the optimal adaptive policy and optimal non-adaptive policy) is at most 3. For the Probemax problem, the best-known approximation ratio is  $1 - \frac{1}{e}$ . Indeed, this can be obtained using the algorithm for stochastic monotone submodular maximization in Asadpour *et al.* [2]. This is also a non-adaptive policy, which implies the adaptivity gap is at most  $\frac{e}{e-1}$ . In this paper, we provide the first PTAS, among all adaptive policies. Note that our policy is indeed adaptive.

**Theorem 1.2.** *There exists a PTAS for the Probemax problem. In other words, for any fixed constant  $\varepsilon > 0$ , there is a polynomial-time approximation algorithm for the Probemax problem that finds a policy with the expected profit at least  $(1 - \varepsilon)\text{OPT}$ , where  $\text{OPT}$  denotes the expected profit of the optimal adaptive policy.*

Let the value  $I_t$  be the maximum among the realized values of the probed items at the time period  $t$ . Using our framework, we have the following system dynamics for Probemax:

$$I_{t+1} = f(I_t, i) = \max\{I_t, X_i\}, \quad g(I_t, i) = 0, \quad \text{and} \quad h(I_{T+1}) = I_{T+1} \quad (1.3)$$

$t = 1, 2, \dots, T$ . Clearly, Assumption 1 (2) and (3) are satisfied. But Assumption 1 (1) is not satisfied because the value space  $\mathcal{V}$  is not of constant size. Hence, we need to discretize the value space and reduce its size to a constant. See Section 4 for more details. If the reward is the summation of top- $k$  values ( $k = O(1)$ ) among the  $m$  realized values, we obtain the ProbeTop- $k$  problem. Our techniques also allow us to derive the following result.

**Theorem 1.3.** *For the ProbeTop- $k$  problem where  $k$  is a constant, there is a polynomial time algorithm that finds an adaptive policy with the expected profit at least  $(1 - \varepsilon)\text{OPT}$ , where  $\text{OPT}$  denotes the expected profit of the optimal adaptive policy.*

### 1.2.2 Committed ProbeTop- $k$ Problem

We are given a set of  $n$  items. Each item  $i \in [n]$  has a value  $X_i$  which is an independent random variable with a known (discrete) distribution  $\pi_i$ . We can *adaptively* probe at most  $m$  items and choose  $k$  values in the committed model, where  $k$  is a constant. In the *committed* model, once we probe an item and observe its value realization, we must make an irrevocable decision whether to choose it or not, *i.e.*, we must either add it to the final chosen set  $C$  immediately or discard it forever.<sup>3</sup> If we add the item to the final chosen set  $C$ , the realized profit is collected. Otherwise, no profit is collected and we are going to probe the next item. Our goal is to design an adaptive probing policy such that the expected value  $\mathbb{E}[\sum_{i \in C} X_i]$  is maximized, where  $C$  is the final chosen set.

<sup>3</sup>In [11, 12], it is called the online decision model.

**Theorem 1.4.** *There is a polynomial time algorithm that finds a committed policy with the expected profit at least  $(1 - \varepsilon)\text{OPT}$  for the committed ProbeTop- $k$  problem, where  $\text{OPT}$  is the expected total profit obtained by the optimal policy.*

Let  $b_i^\theta$  represent the action that we probe item  $i$  with the threshold  $\theta$  (i.e., we choose item  $i$  if  $X_i$  realizes to a value  $s$  such that  $s \geq \theta$ ). Let  $I_t$  be the the number of items that have been chosen at the period time  $t$ . Using our framework, we have following transition dynamics for the ProbeTop- $k$  problem.

$$I_{t+1} = f(I_t, b_i^\theta) = \begin{cases} I_t + 1 & \text{if } X_i \geq \theta, I_t < k, \\ I_t & \text{otherwise;} \end{cases} \quad g(I_t, b_i^\theta) = \begin{cases} X_i & \text{if } X_i \geq \theta, I_t < k, \\ 0 & \text{otherwise;} \end{cases} \quad (1.4)$$

for  $t = 1, 2, \dots, T$ , and  $h(I_{T+1}) = 0$ . Since  $k$  is a constant, Assumption 1 is immediately satisfied. There is one extra requirement for the problem: in any realization path, we can choose at most one action  $b_i^\theta$  from the set  $\mathcal{B}_i = \{b_i^\theta\}_\theta$ . See Section 5 for more details.

### 1.2.3 Committed Pandora’s Box Problem

For Weitzman’s ‘‘Pandora’s box’’ problem [25], we are given  $n$  boxes. For each box  $i \in [n]$ , the probing cost  $c_i$  is deterministic and the value  $X_i$  is an independent random variable with a known (discrete) distribution  $\pi_i$ . Opening a box  $i$  incurs a cost of  $c_i$ . When we open the box  $i$ , its value is realized, which is a sample from the distribution  $\pi_i$ . The goal is to adaptively open a subset  $P \subseteq [n]$  to maximize the expected profit:  $\mathbb{E}[\max_{i \in P}\{X_i\} - \sum_{i \in P} c_i]$ . Weitzman provided an elegant optimal adaptive strategy, which can be computed in polynomial time. Recently, Singla [23] generalized this model to other combinatorial optimization problems such as matching, set cover and so on.

In this paper, we focus on the committed model, which is mentioned in Section 1.2.2. Again, we can *adaptively* open the boxes and choose at most  $k$  values in the committed way, where  $k$  is a constant. Our goal is to design an adaptive policy such that the expected value  $\mathbb{E}[\sum_{i \in C} X_i - \sum_{i \in P} c_i]$  is maximized, where  $C \subseteq P$  is the final chosen set and  $P$  is the set of opened boxes. Although the problem looks like a slight variant of Weitzman’s original problem, it is quite unlikely that we can adapt Weitzman’s argument (or any argument at all) to obtain an optimal policy in polynomial time. When  $k = O(1)$ , we provide the first PTAS for this problem. Note that a PTAS is not known previously even for  $k = 1$ .

**Theorem 1.5.** *When  $k = O(1)$ , there is a polynomial time algorithm that finds a committed policy with the expected value at least  $(1 - \varepsilon)\text{OPT}$  for the committed Pandora’s Box problem.*

Similar to the committed ProbeTop- $k$  problem, let  $b_i^\theta$  represent the action that we open the box  $i$  with threshold  $\theta$ . Let  $I_t$  be the number of boxes that have been chosen at the time period  $t$ . Using our framework, we have following system dynamics for the committed Pandora’s Box problem:

$$I_{t+1} = f(I_t, b_i^\theta) = \begin{cases} I_t + 1 & \text{if } X_i \geq \theta, I_t < k, \\ I_t & \text{otherwise;} \end{cases} \quad g(I_t, b_i^\theta) = \begin{cases} X_i - c_i & \text{if } X_i \geq \theta, I_t < k, \\ -c_i & \text{otherwise;} \end{cases} \quad (1.5)$$

for  $t = 1, 2, \dots, T$ , and  $h(I_{T+1}) = 0$ . Notice that we never take an action  $b_i^\theta$  for a value  $I_t < k$  if  $\mathbb{E}[g(I_t, b_i^\theta)] = \Pr[X_t \geq \theta] \cdot \mathbb{E}[X_i | X_i \geq \theta] - c_i < 0$ . Then Assumption 1 is immediately satisfied. See Section 6 for more details.

### 1.2.4 Stochastic Target Problem

Ilhan *et al.* [16] introduced the following stochastic target problem.<sup>4</sup> In this problem, we are given a predetermined target  $\mathbb{T}$  and a set of  $n$  items. Each item  $i \in [n]$  has a value  $X_i$  which is an independent

<sup>4</sup>[16] called the problem the adaptive stochastic knapsack instead. However, their problem is quite different from the stochastic knapsack problem studied in the theoretical computer science literature. So we use a different name.

random variable with a known (discrete) distribution  $\pi_i$ . Once we decide to insert an item  $i$  into a knapsack, we observe a reward realization  $X_i$  which follows the distribution  $\pi_i$ . We can insert at most  $m$  items into the knapsack and our goal is to design an adaptive policy such that  $\Pr[\sum_{i \in P} X_i \geq \mathbb{T}]$  is maximized, where  $P \subseteq [n]$  is the set of inserted items. For the stochastic target problem, İlhan *et al.* [16] provided some heuristic based on dynamic programming for the special case where the random profit of each item follows a known normal distribution. In this paper, we provide an additive PTAS for the stochastic target problem when the target is relaxed to  $(1 - \varepsilon)\mathbb{T}$ .

**Theorem 1.6.** *There exists an additive PTAS for stochastic target problem if we relax the target to  $(1 - \varepsilon)\mathbb{T}$ . In other words, for any given constant  $\varepsilon > 0$ , there is a polynomial-time approximation algorithm that finds a policy such that the probability of the total rewards exceeding  $(1 - \varepsilon)\mathbb{T}$  is at least  $\text{OPT} - \varepsilon$ , where  $\text{OPT}$  is the resulting probability of an optimal adaptive policy.*

Let the value  $I_t$  be the total profits of the items in the knapsack at time period  $t$ . Using our framework, we have following system dynamics for the stochastic target problem:

$$I_{t+1} = f(I_t, i) = I_t + X_i, \quad g(I_t, i) = 0, \quad \text{and} \quad h(I_{T+1}) = \begin{cases} 1 & \text{if } I_{T+1} \geq \mathbb{T}, \\ 0 & \text{otherwise;} \end{cases} \quad (1.6)$$

for  $t = 1, 2, \dots, T$ . Then Assumption 1 (2,3) is immediately satisfied. But Assumption 1 (1) is not satisfied for that the value space  $\mathcal{V}$  is not of constant size. Hence, we need to discretize the value space and reduce its size to a constant. See Section 7 for more details.

### 1.2.5 Stochastic Blackjack Knapsack

Levin *et al.* [17] introduced the *stochastic blackjack knapsack*. In this problem, we are given a capacity  $\mathbb{C}$  and a set of  $n$  items, each item  $i \in [n]$  has a size  $X_i$  which is an independent random variable with a known distribution  $\pi_i$  and a profit  $p_i$ . We can adaptively insert the items into a knapsack, as long as the capacity constraint is not violated. Our goal is to design an adaptive policy such that the expected total profits of all items inserted is maximized. The key feature here different from classic stochastic knapsack is that we gain zero if overflow, *i.e.*, we will lose the profits of all items inserted already if the total size is larger than the capacity. This extra restriction might induce us to take more conservative policies. Levin *et al.* [17] presented a non-adaptive policy with expected value that is at least  $(\sqrt{2} - 1)^2/2 \approx 1/11.66$  times the expected value of the optimal adaptive policy. Chen *et al.* [7] assumed each size  $X_i$  follows a known exponential distribution and gave an optimal policy for  $n = 2$  based on dynamic programming. In this paper, we provide the first bi-criteria PTAS for the problem.

**Theorem 1.7.** *For any fixed constant  $\varepsilon > 0$ , there is a polynomial-time approximation algorithm for stochastic blackjack knapsack that finds a policy with the expected profit at least  $(1 - \varepsilon)\text{OPT}$ , when the capacity is relaxed to  $(1 + \varepsilon)\mathbb{C}$ , where  $\text{OPT}$  is the expected profit of the optimal adaptive policy.*

Denote  $I_t = (I_{t,1}, I_{t,2})$  and let  $I_{t,1}, I_{t,2}$  be the total sizes and total profits of the items in the knapsack at the time period  $t$  respectively. When we insert an item  $i$  into the knapsack and observe its size realization, say  $s_i$ , we define the system dynamics function to be

$$I_{t+1} = f(I_t, i) = (I_{t,1} + s_i, I_{t,2} + p_i), \quad h(I_{T+1}) = \begin{cases} I_{T+1,2} & \text{if } I_{T+1,1} \leq \mathbb{C}, \\ 0 & \text{otherwise;} \end{cases} \quad (1.7)$$

and  $g(I_t, i) = 0$  for  $t = 1, 2, \dots, T$ . Then Assumption 1 (2,3) is immediately satisfied. But Assumption 1 (1) is not satisfied for that the value space  $\mathcal{V}$  is not of constant size. Hence, we need to discretize the value space and reduce its size to a constant. See Section 8 for more details.

For the case without relaxing the capacity, we can improve the result of 11.66 in [17].

**Theorem 1.8.** *For any  $\varepsilon \geq 0$ , there is a polynomial time algorithm that finds a  $(\frac{1}{8} - \varepsilon)$ -approximate adaptive policy for SBK.*

### 1.3 Related Work

Stochastic dynamic program has been widely studied in computer science and operation research (see, for example, [4, 21]) and has many applications in different fields. It is a natural model for decision making under uncertainty. In 1950s, Richard Bellman [3] introduced the “principle of optimality” which leads to dynamic programming algorithms for solving sequential stochastic optimization problems. However, Bellman’s principle does not immediately lead to efficient algorithms for many problems due to “curse of dimensionality” and the large state space.

There are some constructive frameworks that provide approximation schemes for certain classes of stochastic dynamic programs. Shmoys *et al.* [22] dealt with stochastic linear programs. Halman *et al.* [13, 14, 15] studies stochastic discrete DPs with scalar state and action spaces and designed an FPTAS for their framework. As one of the applications, they used it to solve the stochastic ordered adaptive knapsack problem. As a comparison, in our model, the state space  $\mathcal{S} = \mathcal{V} \times 2^{\mathcal{A}}$  is exponentially large and hence cannot be solved by previous framework.

Stochastic knapsack problem SKP is one of the most well-studied stochastic combinatorial optimization problem. We are given a knapsack of capacity  $\mathbb{C}$ . Each item  $i \in [n]$  has a random value  $X_i$  with a known distribution  $\pi_i$  and a profit  $p_i$ . We can adaptively insert the items to the knapsack, as long as the capacity constraint is not violated. The goal is to maximize the expected total profit of all items inserted. For SKP, Dean *et al.* [9] first provide a constant factor approximation algorithm. Later, Bhalgat *et al.* [6] improved that ratio to  $\frac{3}{8} - \varepsilon$  and gave an algorithm with ratio of  $(1 - \varepsilon)$  by using  $\varepsilon$  extra budget for any given constant  $\varepsilon \geq 0$ . In that paper, the authors first introduced the notion of block adaptive policies, which is crucial for this paper. The best known single-criterion approximation factor is 2 [5, 18, 19].

The Probemax problem and ProbeTop- $k$  problem are special cases of the general stochastic probing framework formulated by Gupta *et al.* [12]. They showed that the adaptivity gap of any stochastic probing problem where the outer constraint is prefix-closed and the inner constraint is an intersection of  $p$  matroids is at most  $O(p^3 \log(np))$ , where  $n$  is the number of items. The Bernoulli version of stochastic probing was introduced in [11], where each item  $i \in U$  has a fixed value  $w_i$  and is “active” with an independent probability  $p_i$ . Gupta *et al.* [11] presented a framework which yields a  $\frac{1}{4(k^{in} + k^{out})}$ -approximation algorithm for the case when  $\mathcal{I}_{in}$  and  $\mathcal{I}_{out}$  are respectively an intersection of  $k^{in}$  and  $k^{out}$  matroids. This ratio was improved to  $\frac{1}{(k^{in} + k^{out})}$  by Adamczyk *et al.* [1] using the iterative randomized rounding approach. Weitzman’s Pandora’s Box is a classical example in which the goal is to find out a single random variable to maximize the utility minus the probing cost. Singla [23] generalized this model to other combinatorial optimization problems such as matching, set cover, facility location, and obtained approximation algorithms.

## 2 Policies and Decision Trees

An instance of stochastic dynamic program is given by  $\mathcal{J} = (\mathcal{V}, \mathcal{A}, f, g, h, T)$ . For each item  $a \in \mathcal{A}$  and values  $I, J \in \mathcal{V}$ , we denote  $\Phi_a(I, J) := \Pr[f(I, a) = J]$  and  $\mathcal{G}_a(I) := \mathbb{E}[g(I, a)]$ . The process of applying a feasible adaptive *policy*  $\sigma$  can be represented as a decision tree  $\mathcal{T}_\sigma$ . Each node  $v$  on  $\mathcal{T}_\sigma$  is labeled by a unique item  $a_v \in \mathcal{A}$ . Before selecting the item  $a_v$ , we denote the corresponding time index, the current value and the set of the remaining available items by  $t_v, I_v$  and  $\mathcal{A}(v)$  respectively. Each node has several children, each corresponding to a different value realization (one possible  $f(I_v, a_v)$ ). Let  $e = (v, u)$  be the  $s$ -th edge emanating from  $s \in \mathcal{V}$  where  $s$  is the realized value. We call  $u$  the  $s$ -child of  $v$ . Thus  $e$  has probability  $\pi_e := \pi_{v,s} = \Phi_{a_v}(I_v, s)$  and weight  $w_e := s$ .

We use  $\mathbb{P}(\sigma)$  to denote the expected profit that the policy  $\sigma$  can obtain. For each node  $v$  on  $\mathcal{T}_\sigma$ , we define  $\mathcal{G}_v := \mathcal{G}_{a_v}(I_v)$ . In order to clearly illustrate the tree structure, we add a dummy node at the end of each root-to-leaf path and set  $\mathcal{G}_v = h(I_v)$  if  $v$  is a dummy node. Then, we recursively define the

expected profit of the subtree  $\mathcal{T}_v$  rooted at  $v$  to be

$$\mathbb{P}(v) = \mathcal{G}_v + \sum_{e=(v,u)} \pi_e \cdot \mathbb{P}(u), \quad (2.1)$$

if  $v$  is an internal node and  $\mathbb{P}(v) = \mathcal{G}_v = h(I_v)$  if  $v$  is a leaf (*i.e.*, the dummy node). The expected profit  $\mathbb{P}(\sigma)$  of the policy  $\sigma$  is simply  $\mathbb{P}(\text{the root of } \mathcal{T}_\sigma)$ . Then, according to Equation (1.2), we have

$$\mathbb{P}(v) \leq \text{DP}_{t_v}(I_v, \mathcal{A}(v)) \leq \text{DP}_1(I_v, \mathcal{A}) \leq \max_{I \in \mathcal{V}} \text{DP}_1(I, \mathcal{A}) = \text{MAX}$$

for each node  $v$ . For a node  $v$ , we say the path from the root to it on  $\mathcal{T}_\sigma$  as the *realization path* of  $v$ , and denote it by  $\mathcal{R}(v)$ . We denote the probability of reaching  $v$  as  $\Phi(v) = \Phi(\mathcal{R}(v)) = \prod_{e \in \mathcal{R}(v)} \pi_e$ . Then, we have

$$\mathbb{P}(\sigma) = \sum_{v \in \mathcal{T}_\sigma} \Phi(v) \cdot \mathcal{G}_v. \quad (2.2)$$

We use  $\text{OPT}$  to denote the expected profit of the optimal adaptive policy. For each node  $v$  on the tree  $\mathcal{T}_\sigma$ , by Assumption 1 (2) that  $f(I_v, a_v) \geq I_v$ , we define  $\mu_v := \Pr[f(I_v, a_v) > I_v] = 1 - \Phi_{a_v}(I_v, I_v)$ . For a set of nodes  $P$ , we define  $\mu(P) := \sum_{v \in P} \mu_v$ .

**Lemma 2.1.** *Given an policy  $\sigma$ , there is a policy  $\sigma'$  with profit at least  $\text{OPT} - O(\varepsilon) \cdot \text{MAX}$  which satisfies that for any realization path  $\mathcal{R}$ ,  $\mu(\mathcal{R}) \leq O(1/\varepsilon)$ , where  $\text{MAX} = \max_{I \in \mathcal{V}} \text{DP}_1(I, \mathcal{A})$ .*

*Proof.* Consider a random realization path  $\mathcal{R} = (v_1, v_2, \dots, v_{T+1})$  generated by  $\sigma$ . Recall in Assumption 1 (1), the value space is  $\mathcal{V} = \{0, 1, \dots, |\mathcal{V}| - 1\}$ . For each node  $v$  on the tree, we define  $y_v := \mathbb{E}[f(I_v, a_v)] - I_v$ , which is larger than

$$I_v \cdot \Pr[f(I_v, a_v) = I_v] + (I_v + 1) \cdot \Pr[f(I_v, a_v) > I_v] - I_v = \Pr[f(I_v, a_v) > I_v] = \mu_v.$$

We now define a sequence of random variables  $\{Y_t\}_{t \in [T+1]}$ :

$$Y_t = I_t - \sum_{i=1}^{t-1} y_{v_i}.$$

This sequence  $\{Y_i\}$  is a martingale: conditioning on current value  $Y_t$ , we have

$$\begin{aligned} \mathbb{E}[Y_{t+1} \mid Y_t] &= \mathbb{E} \left[ I_{t+1} - \sum_{i=1}^t y_{v_i} \mid Y_t \right] \\ &= \mathbb{E} \left[ \left( I_t - \sum_{v=1}^{t-1} y_{v_i} \right) + I_{t+1} - I_t - y_{v_t} \mid Y_t \right] \\ &= Y_t + \mathbb{E}[I_{t+1} \mid Y_t] - I_t - y_{v_t} = Y_t. \end{aligned}$$

The last equation is due to the definition of  $y_{v_t}$ . By the martingale property, we have  $\mathbb{E}[Y_{T+1}] = \mathbb{E}[Y_t] = Y_1 = 0$  for any  $t \in [T]$ . Thus, we have

$$|\mathcal{V}| \geq \mathbb{E}[I_{T+1}] = \mathbb{E} \left[ \sum_{i=1}^T y_{v_i} \right] = \mathbb{E} \left[ \sum_{v \in \mathcal{R}} y_v \right] \geq \mathbb{E}[\mu(\mathcal{R})].$$

Let  $E$  be the set of realization paths  $r$  on the tree for which  $\mu(r) \geq 1/\varepsilon$ . Then, we have  $\mathbb{E}[\mu(\mathcal{R})] \geq \sum_{r \in E} [\Phi(r) \cdot \frac{1}{\varepsilon}]$  which implies that  $\sum_{r \in E} \Phi(r) \leq \varepsilon \cdot \mathbb{E}[\mu(\mathcal{R})] \leq O(\varepsilon)$ , where  $\Phi(r)$  is the probability of passing the path  $r$ . For each path  $r \in E$ , let  $v_r$  be the first node on the path such that  $\mu(\mathcal{R}(v_r)) \geq 1/\varepsilon$ , where  $\mathcal{R}(v_r)$  is the path from the root to the node  $v_r$ . Let  $F$  be the set of such nodes. For the policy



$\sigma$ , we have a truncation on the node  $v_r$  when we reach the node  $v_r$ , *i.e.*, we do not select items (include  $v_r$ ) any more in the new policy  $\sigma'$ . The total profit loss is at most

$$\sum_{v \in F} [\Phi(v) \cdot \mathbb{P}(v)] \leq \text{MAX} \cdot \sum_{r \in E} \Phi(r) \leq O(\varepsilon) \cdot \text{MAX},$$

where  $\text{MAX} = \max_{I \in \mathcal{V}} \text{DP}_1(I, \mathcal{A})$ . □

W.l.o.g, we assume that all (optimal or near optimal) policies  $\sigma$  considered in this paper satisfy that for any realization  $\mathcal{R}$ ,  $\mu(\mathcal{R}) \leq O(1/\varepsilon)$ .

### 3 Block Adaptive Policies

The decision tree corresponding to the optimal policy may be exponentially large and arbitrarily complicated. Now we consider a restrict class of policies, called block-adaptive policy. The concept was first introduced by Bhalgat *et al.* [6] in the context of stochastic knapsack. Our construction is somewhat different from that in [6, 18]. Here, we need to define an order for each block and introduce the notion of approximate block policy.

Formally, a block-adaptive policy  $\hat{\sigma}$  can be thought as a decision tree  $\mathcal{T}_{\hat{\sigma}}$ . Each node on the tree is labeled by a *block* which is a set of items. For a block  $M$ , we choose an arbitrary order  $\varphi$  for the items in the block. According to the order  $\varphi$ , we take the items one by one, until we get a bigger value or all items in the block are taken but the value does not change (recall from Assumption 1 that the value is nondecreasing). Then we visit the child block which corresponds to the realized value. We use  $I_M$  to denote the current value right before taking the items in the block  $M$ . Then for each edge  $e = (M, N)$ , it has probability

$$\pi_e^\varphi = \sum_{a \in M} \left[ \left( \prod_{\varphi_b < \varphi_a} \Phi_b(I_M, I_M) \right) \cdot \Phi_a(I_M, I_N) \right]$$

if  $I_N > I_M$  and  $\pi_e^\varphi = \prod_{a \in M} \Phi_a(I_M, I_M)$  if  $I_N = I_M$ .

Similar to Equation (2.1), for each block  $M$  and an arbitrary order  $\varphi$  for  $M$ , we recursively define the expected profit of the subtree  $\mathcal{T}_M$  rooted at  $M$  to be

$$\mathbb{P}(M) = \mathcal{G}_M^\varphi + \sum_{e=(M,N)} \pi_e^\varphi \cdot \mathbb{P}(N) \quad (3.1)$$

if  $M$  is an internal block and  $\mathbb{P}(M) = h(I_M)$  if  $M$  is a leaf (*i.e.*, the dummy node). Here  $\mathcal{G}_M^\varphi$  is the expected profit we can get from the block which is equal to

$$\mathcal{G}_M^\varphi = \sum_{a \in M} \left[ \left( \prod_{\varphi_b < \varphi_a} \Phi_b(I_M, I_M) \right) \cdot \mathcal{G}_a(I_M) \right].$$

Since the profit  $\mathcal{G}_M^\varphi$  and the probability  $\pi_e^\varphi$  are dependent on the order  $\varphi$  and thus difficult to deal with, we define the approximate block profit and the approximate probability which do not depend on the choice of the specific order  $\varphi$ :

$$\tilde{\mathcal{G}}_M = \sum_{a \in M} \mathcal{G}_a(I_M) \quad \text{and} \quad \tilde{\pi}_e = \sum_{a \in M} \left[ \left( \prod_{b \in M \setminus a} \Phi_b(I_M, I_M) \right) \cdot \Phi_a(I_M, I_N) \right] \quad (3.2)$$

if  $I_N > I_M$  and  $\tilde{\pi}_e = \prod_{a \in M} \Phi_a(I_M, I_M)$  if  $I_N = I_M$ . Then we recursively define the approximate profit

$$\tilde{\mathbb{P}}(M) = \tilde{\mathcal{G}}_M + \sum_{e=(M,N)} \tilde{\pi}_e \cdot \tilde{\mathbb{P}}(N), \quad (3.3)$$

if  $M$  is an internal block and  $\tilde{\mathbb{P}}(M) = \mathbb{P}(M) = h(I_M)$  if  $M$  is a leaf. For each block  $M$ , we define  $\mu(M) := \sum_{a \in M} [1 - \Phi_a(I_M, I_M)]$ . Lemma 3.1 below can be used to bound the gap between the approximate profit and the original profit if the policy satisfies the following property. Then it suffices to consider the approximate profit for a block adaptive policy  $\hat{\sigma}$  in this paper.

(P1) Each block  $M$  with more than one item satisfies that  $\mu(M) \leq \varepsilon^2$ .

**Lemma 3.1.** *For any block-adaptive policy  $\hat{\sigma}$  satisfying Property (P1), we have*

$$(1 + O(\varepsilon^2)) \cdot \tilde{\mathbb{P}}(\hat{\sigma}) \geq \mathbb{P}(\hat{\sigma}) \geq (1 - \varepsilon^2) \cdot \tilde{\mathbb{P}}(\hat{\sigma}).$$

*Proof.* The right hand of this lemma can be proved by induction: for each block  $M$  on the decision tree, we have

$$\mathbb{P}(M) \geq (1 - \varepsilon^2) \cdot \tilde{\mathbb{P}}(M). \quad (3.4)$$

If  $M$  is a leaf, we have  $\mathbb{P}(M) = \tilde{\mathbb{P}}(M)$  which implies that Equation (3.4) holds. For an internal block  $M$ , by Property (P1), we have

$$\mathcal{G}_M^\varphi \geq \left[ \prod_{b \in M} \Phi_b(I_M, I_M) \right] \cdot \sum_{a \in M} \mathcal{G}_a(I_M) \geq \left[ 1 - \sum_{b \in M} (1 - \Phi_b(I_M, I_M)) \right] \cdot \tilde{\mathcal{G}}_M \geq (1 - \varepsilon^2) \cdot \tilde{\mathcal{G}}_M$$

if  $M$  has more than one item and  $\mathcal{G}_M^\varphi = \tilde{\mathcal{G}}_M$  if  $M$  has only one item. For each edge  $e = (M, N)$ , we have  $\pi_e^\varphi \geq \tilde{\pi}_e$ . Then, by induction, we have

$$\begin{aligned} \mathbb{P}(M) &= \mathcal{G}_M^\varphi + \sum_{e=(M,N)} \pi_e^\varphi \cdot \mathbb{P}(N) \\ &\geq (1 - \varepsilon^2) \cdot \tilde{\mathcal{G}}_M + \sum_{e=(M,N)} \tilde{\pi}_e \cdot \left[ (1 - \varepsilon^2) \cdot \tilde{\mathbb{P}}(N) \right] \\ &= (1 - \varepsilon^2) \cdot \tilde{\mathbb{P}}(M). \end{aligned}$$

To prove the left hand of the lemma, we use Equation (2.2):

$$\mathbb{P}(\hat{\sigma}) = \sum_{M \in \mathcal{T}_{\hat{\sigma}}} \Phi(M) \cdot \mathcal{G}_M^\varphi$$

where  $\Phi(M)$  is the probability of reaching the block  $M$ . For each edge  $e = (M, N)$ , if  $I_M = I_N$  or  $M$  has only one item, we have  $\tilde{\pi}_e = \pi_e^\varphi$ . Otherwise, we have

$$\tilde{\pi}_e \geq \left[ \prod_{b \in M} \Phi_b(I_M, I_M) \right] \cdot \sum_{a \in M} \Phi_a(I_M, I_N) \geq (1 - \varepsilon^2) \cdot \sum_{a \in M} \Phi_a(I_M, I_N) \geq (1 - \varepsilon^2) \cdot \pi_e^\varphi.$$

Then, for each block  $M$  and its realization path  $\mathcal{R}(M) = (M_0, M_1, \dots, M_m = M)$ , we have

$$\frac{\tilde{\Phi}(M)}{\Phi(M)} = \prod_{i=0}^{m-1} \frac{\tilde{\pi}_{(M_i, M_{i+1})}}{\pi_{(M_i, M_{i+1})}^\varphi} = \prod_{i: I_{M_i} < I_{M_{i+1}}} \frac{\tilde{\pi}_{(M_i, M_{i+1})}}{\pi_{(M_i, M_{i+1})}^\varphi} \geq (1 - \varepsilon^2)^{|\mathcal{V}|} = 1 - O(\varepsilon^2),$$

where the last inequality holds because the value is nondecreasing and  $|\mathcal{V}| = O(1)$ . Thus we have

$$\tilde{\mathbb{P}}(\hat{\sigma}) = \sum_{M \in \mathcal{T}_{\hat{\sigma}}} \tilde{\Phi}(M) \cdot \tilde{\mathcal{G}}_M \geq \sum_{M \in \mathcal{T}_{\hat{\sigma}}} [(1 - O(\varepsilon^2)) \cdot \Phi(M)] \cdot \mathcal{G}_M \geq (1 - O(\varepsilon^2)) \cdot \mathbb{P}(\hat{\sigma}).$$

□

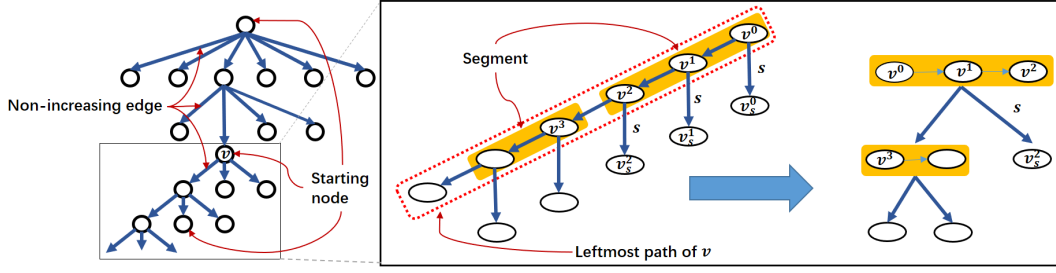


Figure 1: Decision tree and block policy

### 3.1 Constructing a Block Adaptive Policy

In this section, we show that there exists a block-adaptive policy that approximates the optimal adaptive policy. In order to prove this, from an optimal (or nearly optimal) adaptive policy  $\sigma$ , we construct a block adaptive policy  $\hat{\sigma}$  which satisfies certain nice properties and can obtain almost as much profit as  $\sigma$  does. Thus it is sufficient to restrict our search to the block-adaptive policies. The construction is similar to that in [18].

**Lemma 3.2.** *An optimal policy  $\sigma$  can be transformed into a block adaptive policy  $\hat{\sigma}$  with approximate expected profit  $\tilde{\mathbb{P}}(\hat{\sigma})$  at least  $\text{OPT} - O(\varepsilon) \cdot \text{MAX}$ . Moreover, the block-adaptive policy  $\hat{\sigma}$  satisfies Property (P1) and (P2):*

(P1) *Each block  $M$  with more than one item satisfies that  $\mu(M) \leq \varepsilon^2$ .*

(P2) *There are at most  $O(\varepsilon^{-3})$  blocks on any root-to-leaf path on the decision tree.*

*Proof.* For a node  $v$  on the decision tree  $\mathcal{T}_\sigma$  and a value  $s \in \mathcal{V}$ , we use  $v_s$  to denote the  $s$ -child of  $v$ , which is the child of  $v$  corresponding to the realized value  $s$ . We say an edge  $e_{v,u}$  is *non-increasing* if  $I_v = I_u$  and define the *leftmost path* of  $v$  to be the realization path which starts at  $v$ , ends at a leaf, and consists of only the non-increasing edges.

We say a node  $v$  is a *starting node* if  $v$  is the root or  $v$  corresponds to an increasing value of its parent  $v'$  (i.e.,  $I_v > I_{v'}$ ). For each starting node  $v$ , we greedily partition the leftmost path of  $v$  into several segments such that for any two nodes  $u, w$  in the same segment  $M$  and for any value  $s \in \mathcal{V}$ , we have

$$|\mathbb{P}(u_s) - \mathbb{P}(w_s)| \leq \varepsilon^2 \cdot \text{MAX} \text{ and } \mu(M) \leq \varepsilon^2. \quad (3.5)$$

Since  $\mu(\mathcal{R})$  is at most  $O(1/\varepsilon)$  for each root-to-leaf path  $\mathcal{R}$  by Lemma 2.1, the second inequality in (3.5) can yield at most  $O(\varepsilon^{-3})$  blocks. Now focus on the first inequality in (3.5). Fix a particular leftmost path  $\mathcal{R}^v = (v^0, v^1, \dots, v^m)$  from a starting node  $v (v = v^0)$  on  $\mathcal{T}_\sigma$ . For each value  $s \in \mathcal{V}$ , we have

$$\text{MAX} \geq \text{DP}_1(s, \mathcal{A}) \geq \mathbb{P}(v_s^0) \geq \mathbb{P}(v_s^1) \geq \dots \geq \mathbb{P}(v_s^m) \geq 0.$$

Otherwise, replacing the subtree  $T_{v_s^i}$  with  $T_{v_s^j}$  increases the profit of the policy  $\sigma$  for some  $i < j \leq m$  if  $\mathbb{P}(v_s^i) < \mathbb{P}(v_s^j)$ . Thus, for each particular size  $s \in \mathcal{V}$ , we could cut the path  $\mathcal{R}^v$  at most  $\varepsilon^{-2}$  times. Since  $|\mathcal{V}| = O(1)$ , we have at most  $O(\varepsilon^{-2})$  segments on the leftmost path  $\mathcal{R}^v$ . Now, fix a particular root-to-leaf path. Since the value is nondecreasing by Assumption 1 (2), there are at most  $|\mathcal{V}| = O(1)$  starting nodes on the path. Thus the first inequality in (3.5) can yield at most  $O(\varepsilon^{-2})$  segments on the root-to-leaf path. In total, there are at most  $O(\varepsilon^{-3})$  segments on any root-to-leaf path on the decision tree.

Now, we are ready to describe the algorithm, which takes a policy  $\sigma$  as input and outputs a block adaptive policy  $\hat{\sigma}$ . For each node  $v$ , we denote its segment  $\text{seg}(v)$  and use  $l(v)$  to denote the last node in  $\text{seg}(v)$ . In Algorithm 1, we can see that the set of items which the policy  $\hat{\sigma}$  attempts to take

---

**Algorithm 1** A policy  $\hat{\sigma}$ 


---

**Input:** A policy  $\sigma$ .

- 1: We start at the root of  $\mathcal{T}_\sigma$ .
  - 2: **repeat**
  - 3:   Suppose we are at node  $v$  on  $\mathcal{T}_\sigma$ . Take the items in  $\text{seg}(v)$  one by one in the original order (the order of items in policy  $\sigma$ ) until some node  $u$  makes a transition to an increasing value, say  $s$ .
  - 4:   Visit the node  $l(v)_s$ , the  $s$ -child of  $l(v)$  (*i.e.*, the last node of  $\text{seg}(v)$ ).
  - 5:   If all items in  $\text{seg}(v)$  have been taken and the value does not change, visit  $l(v)_{I_v}$ .
  - 6: **until** A leaf on  $\mathcal{T}_\sigma$  is reached.
- 

always corresponds to some realization path in the original policy  $\sigma$ . Property (P1) and (P2) hold immediately following from the partition argument. Now we show that the expected profit  $\mathbb{P}(\hat{\sigma})$  that the new policy  $\hat{\sigma}$  can obtain is at least  $\text{OPT} - O(\varepsilon^2) \cdot \text{MAX}$ .

Our algorithm deviates the policy  $\sigma$  when the first time a node  $u$  in the segment  $\text{seg}(v)$  which makes a transition to an increasing value, say  $s$ . In this case,  $\sigma$  would visit  $u_s$ , the  $s$ -child of  $u$  and follows  $\mathcal{T}_{u_s}$  from then on. But our algorithm visits  $l(v)_s$ , the  $s$ -child of  $l(v)$  (*i.e.*, the last node of  $\text{seg}(v)$ ), and follows  $\mathcal{T}_{l(v)_s}$ . The expected profit gap in each such event can be bounded by

$$\mathbb{P}(u_s) - \mathbb{P}(l(v)_s) \leq \varepsilon^2 \cdot \text{MAX},$$

due to the first inequality in Equation (3.5). Suppose  $\sigma$  pays such a profit loss, and switches to visit  $l(v)_s$ . Then,  $\sigma$  and our algorithm always stay at the same node. Note that there are at most  $|\mathcal{V}| = O(1)$  starting nodes on any root-to-leaf path. Thus  $\sigma$  pays at most  $O(1)$  times in any realization. Therefore, the total profit loss is at most  $O(\varepsilon^2) \cdot \text{MAX}$ . By Lemma 3.1, we have

$$\tilde{\mathbb{P}}(\hat{\sigma}) \geq (1 - O(\varepsilon^2)) \cdot \mathbb{P}(\hat{\sigma}) \geq (1 - O(\varepsilon^2)) \cdot (\text{OPT} - O(\varepsilon^2) \cdot \text{MAX}) \geq \text{OPT} - O(\varepsilon) \cdot \text{MAX}.$$

□

### 3.2 Enumerating Signatures

To search for the (nearly) optimal block-adaptive policy, we want to enumerate all possible structures of the block decision tree. Fixing the topology of the decision tree, we need to decide the subset of items to place in each block. To do this, we define the *signature* such that two subsets with the same signature have approximately the same profit distribution. Then, we can enumerate the signatures of all blocks in polynomial time and find a nearly optimal block-adaptive policy. Formally, for an item  $a \in \mathcal{A}$  and a value  $I \in \mathcal{V} = (0, 1, \dots, |\mathcal{V}| - 1)$ , we define the *signature* of  $a$  on  $I$  to be the following vector

$$\text{Sg}_I(a) = (\bar{\Phi}_a(I, 0), \bar{\Phi}_a(I, 1), \dots, \bar{\Phi}_a(I, |\mathcal{V}| - 1), \bar{\mathcal{G}}_a(I)),$$

where

$$\bar{\Phi}_a(I, J) = \left\lfloor \Phi_a(I, J) \cdot \frac{n}{\varepsilon^4} \right\rfloor \cdot \frac{\varepsilon^4}{n} \quad \text{and} \quad \bar{\mathcal{G}}_a(I) = \left\lfloor \mathcal{G}_a(I) \cdot \frac{n}{\varepsilon^4 \text{MAX}} \right\rfloor \cdot \frac{\varepsilon^4 \text{MAX}}{n}$$

for any  $J \in \mathcal{V}$ .<sup>5</sup> For a block  $M$  of items, we define the *signature* of  $M$  on  $I$  to be

$$\text{Sg}_I(M) = \sum_{a \in M} \text{Sg}_I(a).$$

**Lemma 3.3.** *Consider two decision trees  $\mathcal{T}_1, \mathcal{T}_2$  corresponding to block-adaptive policies with the same topology (i.e.,  $\mathcal{T}_1$  and  $\mathcal{T}_2$  are isomorphic) and the two block adaptive policies satisfy Property (P1) and (P2). If for each block  $M_1$  on  $\mathcal{T}_1$ , the block  $M_2$  at the corresponding position on  $\mathcal{T}_2$  satisfies that  $\text{Sg}_I(M_1) = \text{Sg}_I(M_2)$  where  $I = I_{M_1} = I_{M_2}$ , then  $|\tilde{\mathbb{P}}(\mathcal{T}_1) - \tilde{\mathbb{P}}(\mathcal{T}_2)| \leq O(\varepsilon) \cdot \text{MAX}$ .*

---

<sup>5</sup>If  $\text{MAX} = \max_{I \in \mathcal{V}} \text{DP}_1(I, \mathcal{A})$  is unknown, for some several concrete problems (*e.g.*, Probemax), we can get a constant approximation result for MAX, which is sufficient for our purpose. In general, we can guess a constant approximation result for MAX using binary search.

*Proof.* We focus on the case when  $M$  has more than one item. Recall that for each  $e = (M, N)$ , we have

$$\tilde{\pi}_e = \sum_{a \in M} \left[ \left( \prod_{b \in M \setminus a} \Phi_b(I_M, I_M) \right) \cdot \Phi_a(I_M, I_N) \right]$$

if  $I_N > I_M$  and  $\tilde{\pi}_e = \prod_{a \in M} \Phi_a(I_M, I_M)$  if  $I_N = I_M$ . For simplicity, we use  $(I, J)$  to replace  $(I_M, I_N)$  if the context is clear, and write  $\tilde{\pi}_e$  as  $\pi_M^I$  if  $J = I$  and  $\pi_M^J$  if  $J > I$ .

Fixing a block  $M$ , for each item  $a \in M$ , we define  $\mu_a := \Pr[f(I, a) > I]$ . By Property (P1) that  $\mu(M) = \sum_{a \in M} [1 - \Phi_a(I, I)] = \sum_{a \in M} \mu_a \leq \varepsilon^2$ , we have

$$\pi_M^I = \prod_{a \in M} (1 - \mu_a) \leq \left( 1 - \frac{\sum_{a \in M} \mu_a}{|M|} \right)^{|M|} \leq \exp(-\mu(M)) \leq 1 - \mu(M) + \mu(M)^2 \leq 1 - \mu(M) + \varepsilon^4$$

and  $\pi_M^I = \prod_{a \in M} (1 - \mu_a) \geq 1 - \sum_{a \in M} \mu_a = 1 - \mu(M)$ . Since  $\sum_{a \in M} \Phi_a(I, J) \leq \sum_{a \in M} \mu_a$  for any  $J > I$ , we have

$$\pi_M^J = \left[ \prod_{b \in M} \Phi_b(I, I) \right] \cdot \left[ \sum_{a \in M} \frac{\Phi_a(I, J)}{\Phi_a(I, I)} \right] \geq (1 - \varepsilon^2) \cdot \sum_{a \in M} \Phi_a(I, J) \geq \sum_{a \in M} \Phi_a(I, J) - \varepsilon^4.$$

It is straightforward to verify the following property when  $M$  has only one item:

$$\pi_M^I = 1 - \mu(M) \text{ and } \pi_M^J = \sum_{a \in M} \Phi_a(I, J) \text{ for any } J > I.$$

Let  $M_1, M_2$  be the root blocks of  $\mathcal{T}_1, \mathcal{T}_2$  respectively. Since  $\text{Sg}_I(M_1) = \text{Sg}_I(M_2)$ , we have that

$$\left| \sum_{a \in M_1} \Phi_a(I, J) - \sum_{a \in M_2} \Phi_a(I, J) \right| \leq \varepsilon^4,$$

for any  $J \in \mathcal{V}$ . Then, we have

$$\pi_{M_1}^I - \pi_{M_2}^I \leq 1 - \mu(M_1) + \varepsilon^4 - (1 - \mu(M_2)) = (\mu(M_2) - \mu(M_1)) + \varepsilon^4 = O(\varepsilon^4).$$

and for any  $J > I$

$$\pi_{M_1}^J - \pi_{M_2}^J \leq \sum_{a \in M_1} \Phi_a(I, J) - \sum_{a \in M_2} \Phi_a(I, J) + \varepsilon^4 = O(\varepsilon^4).$$

On the tree  $\mathcal{T}_1$ , we replace  $M_1$  with  $M_2$ . For each  $s \in \mathcal{V}$ , we use  $M_s$  to denote the  $s$ -child of block  $M_1$  on  $\mathcal{T}_1$ . Then we have

$$\begin{aligned} \tilde{\mathbb{P}}(M_1) - \tilde{\mathbb{P}}(M_2) &= \left( \tilde{\mathcal{G}}_{M_1} - \tilde{\mathcal{G}}_{M_2} \right) + \sum_{s \in \mathcal{V}} \tilde{\mathbb{P}}(M_s) \cdot (\pi_{M_1}^s - \pi_{M_2}^s) \\ &\leq \varepsilon^4 \cdot \text{MAX} + O(\varepsilon^4) \cdot \text{MAX} \\ &= O(\varepsilon^4) \cdot \text{MAX}. \end{aligned}$$

We replace all the blocks on  $\mathcal{T}_1$  by the corresponding blocks on  $\mathcal{T}_2$  one by one from the root to leaf. The total profit loss is at most  $\sum_{M \in \mathcal{T}_2} [\Phi(M) \cdot O(\varepsilon^4) \cdot \text{MAX}] \leq O(\varepsilon) \text{MAX}$ , where  $\Phi(M)$  is the probability of reaching  $M$ . The inequality holds because the depth of  $\mathcal{T}_2$  is at most  $O(\varepsilon^{-3})$  by Property (P2), which implies that  $\sum_{M \in \mathcal{T}_2} \Phi(M) \leq O(\varepsilon^{-3})$ .  $\square$

Since  $|V| = O(1)$ , the number of possible signatures for a block is  $O((n/\varepsilon^4)^{|V|}) = n^{O(1)}$ , which is a polynomial of  $n$ . By Lemma 3.2, for any block decision tree  $\mathcal{T}$ , there are at most  $(|V|)^{O(\varepsilon^{-3})} = 2^{O(\varepsilon^{-3})}$  blocks on the tree which is a constant.

### 3.3 Finding a Nearly Optimal Block-adaptive Policy

In this section, we find a nearly optimal block-adaptive policy and prove Theorem 1.1. To do this, we enumerate over all topologies of the decision trees along with all possible signatures for each block. This can be done by a standard dynamic programming.

Consider a given tree topology  $\mathcal{T}$ . A configuration  $\mathbf{C}$  is a set of signatures each corresponding to a block. Let  $t_1$  and  $t_2$  be the number of paths and blocks on  $\mathcal{T}$  respectively. We define a vector  $\mathbf{CA} = (u_1, u_2, \dots, u_{t_1})$  where  $u_j$  is the upper bound of the number of items on the  $j$ th path. For each given  $i \in [n]$ ,  $\mathbf{C}$  and  $\mathbf{CA}$ , let  $\mathcal{M}(i, \mathbf{C}, \mathbf{CA}) = 1$  indicate that we can reach the configuration  $\mathbf{C}$  using a subset of items  $\{a_1, \dots, a_i\}$  such that the total number of items on each path  $j$  is no more than  $u_j$  and 0 otherwise. Set  $\mathcal{M}(0, \mathbf{0}, \mathbf{0}) = 1$  and we compute  $\mathcal{M}(i, \mathbf{C}, \mathbf{CA})$  in an lexicographically increasing order of  $(i, \mathbf{C}, \mathbf{CA})$  as follows:

$$\mathcal{M}(i, \mathbf{C}, \mathbf{CA}) = \max \left\{ \mathcal{M}(i-1, \mathbf{C}, \mathbf{CA}), \mathcal{M}(i-1, \mathbf{C}', \mathbf{CA}') \right\} \quad (3.6)$$

Now, we explain the above recursion as follows. In each step, we should decide how to place the item  $a_i$  on the tree  $\mathcal{T}$ . Notice that there are at most  $t_2 = (|\mathcal{V}|)^{O(\varepsilon^{-3})} = 2^{O(\varepsilon^{-3})}$  blocks and therefore at most  $2^{t_2}$  possible placements of item  $a_i$  and each placement is called *feasible* if there are no two blocks on which we place the item  $a_i$  have an ancestor-descendant relation. For a feasible placement of  $a_i$ , we subtract  $\text{Sg}(a_i)$  from each entry in  $\mathbf{C}$  corresponding to the block we place  $a_i$  and subtract 1 from  $\mathbf{CA}$  on each entry corresponding to a path including  $a_i$ , and in this way we get the resultant configuration  $\mathbf{C}'$  and  $\mathbf{CA}'$  respectively. Hence, the max is over all possible such  $\mathbf{C}', \mathbf{CA}'$ .

We have shown that the total number of all possible configurations on  $\mathcal{T}$  is  $n^{t_2}$ . The total number of vectors  $\mathbf{CA}$  is  $T^{t_1} \leq n^{t_1} \leq n^{t_2} = n^{t_2}$  where  $T$  is the number of rounds. For each given  $(i, \mathbf{C}, \mathbf{CA})$ , the computation takes a constant time  $O(2^{t_2})$ . Thus we claim for a given tree topology, finding the optimal configuration can be done within  $O(n^{2^{O(\varepsilon^{-3})}})$  time .

*The proof of Theorem 1.1.* Suppose  $\sigma^*$  is the optimal policy with expected profit  $\mathbb{P}(\sigma^*) = \text{OPT}$ . We use the above dynamic programming to find a nearly optimal block adaptive policy  $\sigma$ . By Lemma 3.2, there exists a block adaptive policy  $\hat{\sigma}$  such that

$$\tilde{\mathbb{P}}(\hat{\sigma}) \geq \text{OPT} - O(\varepsilon)\text{MAX}.$$

Since the configuration of  $\hat{\sigma}$  is enumerated at some step of the algorithm, our dynamic programming is able to find a block adaptive policy  $\sigma$  with the same configuration (the same tree topology and the same signatures for corresponding blocks). By Lemma 3.3, we have

$$\tilde{\mathbb{P}}(\sigma) \geq \tilde{\mathbb{P}}(\hat{\sigma}) - O(\varepsilon)\text{MAX} \geq \text{OPT} - O(\varepsilon)\text{MAX}.$$

By Lemma 3.1, we have  $\mathbb{P}(\sigma) \geq (1 - \varepsilon^2) \cdot \tilde{\mathbb{P}}(\sigma) \geq \text{OPT} - O(\varepsilon)\text{MAX}$ . Hence, the proof of Theorem 1.1 is completed.  $\square$

## 4 Probemax Problem

In this section, we demonstrate the application of our framework to the Probemax problem. Define the value set  $S = \bigcup_{i \in [n]} S_i$  where  $S_i$  is the support of the random variable  $X_i$  and the item set  $\mathcal{A} = \{1, 2, \dots, n\}$ . Let the value  $I_t$  be the maximum among the realized values of the probed items at the time period  $t$ . Thus, we begin with the initial value  $I_1 = 0$ . Since we can probe at most  $m$  items, we set the number of rounds to be  $T = m$ . When we probe an item  $i$  and observe its value realization, say  $X_i$ , we have the system dynamic functions

$$I_{t+1} = f(I_t, i) = \max\{I_t, X_i\}, \quad g(I_t, i) = 0, \quad \text{and} \quad h(I_{T+1}) = I_{T+1} \quad (4.1)$$

for  $I_t \in S$  and  $t = 1, 2, \dots, T$ . Assumption 1 (2,3) is immediately satisfied. But Assumption 1 (1) is not satisfied because the value space  $S$  is not of constant size. Hence, we need discretization.

## 4.1 Discretization

Now, we need to discretize the value space, using parameter  $\varepsilon$ . We start with a constant factor approximate solution  $\widetilde{\text{OPT}}$  for the Probemax problem with  $\text{OPT} \geq \widetilde{\text{OPT}} \geq (1 - 1/e)^2 \text{OPT}$  (this can be obtained by a simple greedy algorithm See e.g., Appendix C of [8]). Let  $X$  be a discrete random variable with a support  $S = (s_1, s_2, \dots, s_l)$  and  $p_{s_i} = \Pr[X = s_i]$ . Let  $\theta = \frac{\widetilde{\text{OPT}}}{\varepsilon}$  be a threshold. For “large” size  $s_i$ , i.e.,  $s_i \geq \theta$ , set  $D_X(s_i) = \theta$ . For “small” size  $s_i$ , i.e.,  $s_i < \theta$ , set  $D_X(s_i) = \lfloor \frac{s_i}{\varepsilon \widetilde{\text{OPT}}} \rfloor \varepsilon \widetilde{\text{OPT}}$ . We use  $\mathcal{V} = \{0, \varepsilon \widetilde{\text{OPT}}, \dots, \widetilde{\text{OPT}}/\varepsilon\}$  to denote the discretized support. Now, we describe the discretized random variable  $\widetilde{X}$  with the support  $\mathcal{V}$ . For “large” size, we set

$$\tilde{p}_\theta = \Pr[\widetilde{X} = \theta] = \Pr[X \geq \theta] \cdot \frac{\mathbb{E}[X \mid X \geq \theta]}{\theta}. \quad (4.2)$$

Under the constraint that the sum of probabilities remains 1, for “small” size  $d \in \mathcal{V} \setminus \{\theta\}$ , we scale down the probability by setting

$$\tilde{p}_d = \Pr[\widetilde{X} = d] = \frac{1 - \Pr[\widetilde{X} = \theta]}{\Pr[X < \theta]} \cdot \left( \sum_{s \in S, D_X(s)=d} \Pr[X = s] \right) \leq \sum_{s \in S, D_X(s)=d} \Pr[X = s]. \quad (4.3)$$

Although the above discretization is quite natural, there are some technical details. We know how to solve the problem for the discretized random variables supported on  $\mathcal{V}$  but the realized values are in  $S$ . Hence, we need to introduce the notion of *canonical policies* (the notion was introduced in Bhargat et al. [6] for stochastic knapsack). The policy makes decisions based on the discretized sizes of variables, not their true size. More precisely, when the canonical policy  $\tilde{\sigma}$  probes an item  $X$  which realizes to  $s \in S$ , the policy makes decisions based on discretized size  $D_X(s)$ . In this following lemma, we show it suffices to only consider canonical policies. We use  $\mathbb{P}(\sigma, \pi)$  to denote the expected profit that the policy  $\sigma$  can obtain with the given distribution  $\pi$ .

**Lemma 4.1.** *Let  $\pi = \{\pi_i\}$  be the set of distributions of random variables and  $\tilde{\pi}$  be the discretized version of  $\pi$ . Then, we have:*

1. *For any policy  $\sigma$ , there exists a (canonical) policy  $\tilde{\sigma}$  such that*

$$\mathbb{P}(\tilde{\sigma}, \tilde{\pi}) \geq (1 - O(\varepsilon))\mathbb{P}(\sigma, \pi) - O(\varepsilon)\text{OPT};$$

2. *For any canonical policy  $\tilde{\sigma}$ ,*

$$\mathbb{P}(\tilde{\sigma}, \pi) \geq \mathbb{P}(\tilde{\sigma}, \tilde{\pi}).$$

The proof of the lemma can be found in Appendix A.

*The proof of Theorem 1.2.* Suppose  $\sigma^*$  is the optimal policy with expected profit  $\mathbb{P}(\sigma^*, \pi) = \text{OPT}$ . Given an instance  $\pi$ , we compute the discretized distribution  $\tilde{\pi}$ . By Lemma 4.1 (1), there exists a canonical policy  $\tilde{\sigma}^*$  such that

$$\mathbb{P}(\tilde{\sigma}^*, \tilde{\pi}) \geq (1 - O(\varepsilon)) \cdot \mathbb{P}(\sigma^*, \pi) - O(\varepsilon)\text{OPT} = (1 - O(\varepsilon))\text{OPT}.$$

Now, we present a stochastic dynamic program for the Probemax problem with the discretized distribution  $\tilde{\pi}$ . Define the value set  $\mathcal{V} = \{0, \varepsilon \widetilde{\text{OPT}}, \dots, \widetilde{\text{OPT}}/\varepsilon\}$  and the item set  $\mathcal{A} = \{1, 2, \dots, n\}$ , and set  $T = m$  and  $I_1 = 0$ . When we probe an item  $i$  to observe its value realization, say  $X_i$ , we define the system dynamic functions to be

$$I_{t+1} = f(I_t, i) = \max\{I_t, X_i\}, \quad g(I_t, i) = 0, \quad \text{and} \quad h(I_{T+1}) = I_{T+1} \quad (4.4)$$

for  $I_t \in \mathcal{V}$  and  $t = 1, 2, \dots, T$ . Then Assumption 1 is immediately satisfied. By Theorem 1.1, we can find a policy  $\sigma$  with profit at least

$$\text{OPT}_d - O(\varepsilon^2) \cdot \text{MAX},$$

where  $\text{OPT}_d$  denotes the expected profit of the optimal policy for the discretized version  $\tilde{\pi}$  and  $\text{MAX} = \max_{I \in \mathcal{V}} \text{DP}_1(I, \mathcal{A}) = \text{DP}_1(\widetilde{\text{OPT}}/\varepsilon, \mathcal{A}) = \widetilde{\text{OPT}}/\varepsilon$ . We can see that  $\text{OPT}_d \geq \mathbb{P}(\tilde{\sigma}^*, \tilde{\pi}) \geq (1 - O(\varepsilon))\text{OPT}$ . Thus, by Lemma 4.1 (2), we have

$$\mathbb{P}(\sigma, \pi) \geq \mathbb{P}(\sigma, \tilde{\pi}) \geq \text{OPT}_d - O(\varepsilon^2)\text{MAX} \geq (1 - O(\varepsilon))\text{OPT} - O(\varepsilon)\text{OPT} = (1 - O(\varepsilon))\text{OPT},$$

which completes the proof.  $\square$

## 4.2 ProbeTop- $k$ Problem

In this section, we consider the ProbeTop- $k$  problem where the reward is the summation of top- $k$  values and  $k$  is a constant.

**Theorem 4.2.** *There exists a PTAS for the ProbeTop- $k$  problem. In other words, for any fixed constant  $\varepsilon > 0$ , there is a polynomial-time approximation algorithm for the ProbeTop- $k$  problem that finds a policy with the expected value at least  $(1 - \varepsilon)\text{OPT}$ .*

In this case,  $I_t$  is a vector of the top- $k$  values among the realized value of the probed items at the time period  $t$ . Thus, we begin with the initial vector  $I_1 = \{0\}^k$ . When we probe an item  $i$  and observe its value realization  $X_i$ , we update the vector by

$$I_{t+1} = \{I_t + X_i\} \setminus \min\{I_t, X_i\}.$$

We set  $g(I_t, i) = 0$  and  $h(I_{T+1}) = \text{sum}(I_{T+1})$ . Assumption 1 (2,3) is immediately satisfied. Then We also need the discretization to satisfy the Assumption 1 (1). For Lemma 4.1, we make a small change as shown in Lemma 4.3. The proof of the lemma can be found in Appendix B. Thus, we can prove Theorem 4.2 which is essentially the same as the proof of Theorem 1.2 and we omit it here.

**Lemma 4.3.** *Let  $\pi = \{\pi_i\}$  be the set of distributions of random variables and  $\tilde{\pi}$  be the discretized version of  $\pi$ . Then, we have:*

1. *For any policy  $\sigma$ , there exists a canonical policy  $\tilde{\sigma}$  such that*

$$\mathbb{P}(\tilde{\sigma}, \tilde{\pi}) \geq (1 - O(\varepsilon))\mathbb{P}(\sigma, \pi) - O(\varepsilon)\text{OPT};$$

2. *For any canonical policy  $\tilde{\sigma}$ ,*

$$\mathbb{P}(\tilde{\sigma}, \pi) \geq \mathbb{P}(\tilde{\sigma}, \tilde{\pi}) - O(\varepsilon)\text{OPT}.$$

## 5 Committed ProbeTop- $k$ Problem

In this section, we prove Theorem 1.4, *i.e.*, obtaining a PTAS for the committed ProbeTop- $k$ . In the committed model, once we probe an item and observe its value realization, we are committed to making an irrevocable decision immediately whether to choose it or not. If we add the item to the final chosen set  $C$ , the realized profit is collected. Otherwise, no profit is collected and we are going to probe the next item.

Let  $\sigma^*$  be the optimal committed policy. Suppose  $\sigma^*$  is going to probe the item  $i$  and choose the item  $i$  if  $X_i$  realizes to a value  $\theta \in S_i$ , where  $S_i$  is the support of the random variable  $X_i$ . Then  $\sigma^*$  would choose the item  $i$  if  $X_i$  realizes to a larger value  $s \geq \theta$ . We call  $\theta$  threshold for the item  $i$ . Thus the committed policy  $\sigma^*$  for the committed ProbeTop- $k$  problem can be represented as a decision tree  $\mathcal{T}_{\sigma^*}$ . Every node  $v$  is labeled by an unique item  $a_v$  and a threshold  $\theta(v)$ , which means the policy chooses the item  $a_v$  if  $X_v$  realizes to a size  $s \geq \theta(v)$ , and otherwise rejects it.



Now, we present a stochastic dynamic program for this problem. For each item  $i$ , we create a set of actions  $\mathcal{B}_i = \{b_i^\theta\}_\theta$ , where  $b_i^\theta$  represents the action that we probe item  $i$  with the threshold  $\theta$ . Since we assume discrete distribution (given explicitly as the input), there are at most a polynomial number of thresholds. Hence the set of action  $\mathcal{A} = \cup_{i \in [n]} \mathcal{B}_i$  is bounded by a polynomial. The only requirement is that at most one action from  $\mathcal{B}_i$  can be selected.

Let  $I_t$  be the the number of items that have been chosen at the period time  $t$ . Then we set  $\mathcal{V} = \{0, 1, \dots, k\}$ ,  $I_1 = 0$ . Since we can probe at most  $m$  items, we set  $T = m$ . When we select an action  $b_i^\theta$  to probe the item  $i$  and observe its value realization, say  $X_i$ , we define the system dynamic functions to be

$$I_{t+1} = f(I_t, b_i^\theta) = \begin{cases} I_t + 1 & \text{if } X_i \geq \theta, I_t < k, \\ I_t & \text{otherwise;} \end{cases} \quad g(I_t, b_i^\theta) = \begin{cases} X_i & \text{if } X_i \geq \theta, I_t < k, \\ 0 & \text{otherwise;} \end{cases} \quad (5.1)$$

for  $I_t \in \mathcal{V}$  and  $t = 1, 2, \dots, T$ , and  $h(I_{T+1}) = 0$ . Since  $k$  is a constant, Assumption 1 is immediately satisfied. However, in this case, we cannot directly use Theorem 1.1, due to the extra requirement that at most one action from each  $\mathcal{B}_i$  can be selected. In this case, we need to slightly modify the dynamic program in Section 3.3 to satisfy the requirement. To compute  $\mathcal{M}(i, \mathcal{C}, \mathcal{CA})$ , once we decide the position of the item  $i$ , we need to choose a threshold for the item. Since there are at most a polynomial number of thresholds, it can be computed at polynomial time. Hence, again, we can find a policy  $\sigma$  with profit at least

$$\text{OPT} - O(\varepsilon) \cdot \text{MAX} = (1 - O(\varepsilon)) \text{OPT},$$

where  $\text{OPT}$  denotes the expected profit of the optimal policy and  $\text{MAX} = \max_{I \in \mathcal{V}} \text{DP}_1(I, \mathcal{A}) = \text{DP}_1(0, \mathcal{V}) = \text{OPT}$ .

## 6 Committed Pandora's Box Problem

In this section, we obtain a PTAS for the committed Pandora's Box problem. This can be proved by an analogous argument to Theorem 1.4 in Section 5. Similarly, for each box  $i$ , we create a set of actions  $\mathcal{B}_i = \{b_i^\theta\}$ , where  $b_i^\theta$  represents the action that we open the box  $i$  with threshold  $\theta$ . Let  $I_t$  be the number of boxes that have been chosen at the time period  $t$ . Then we set  $\mathcal{A} = \cup_{i \in [n]} \mathcal{B}_i$ ,  $\mathcal{V} = \{0, 1, \dots, k\}$ ,  $T = n$  and  $I_1 = 0$ . When we select an action  $b_i^\theta$  to open the box  $i$  and observe its value realization, say  $X_i$ , we define the system dynamic functions to be

$$I_{t+1} = f(I_t, b_i^\theta) = \begin{cases} I_t + 1 & \text{if } X_i \geq \theta, I_t < k, \\ I_t & \text{otherwise;} \end{cases} \quad g(I_t, b_i^\theta) = \begin{cases} X_i - c_i & \text{if } X_i \geq \theta, I_t < k, \\ -c_i & \text{otherwise;} \end{cases} \quad (6.1)$$

for  $I_t \in \mathcal{V}$  and  $t = 1, 2, \dots, T$ , and  $h(I_{T+1}) = 0$ . Notice that we never take an action  $b_i^\theta$  for a value  $I_t < k$  if  $\mathbb{E}[g(I_t, b_i^\theta)] = \Pr[X_t \geq \theta] \cdot \mathbb{E}[X_i | X_i \geq \theta] - c_i < 0$ . Then Assumption 1 is immediately satisfied. Similar to the Committed ProbeTop- $k$  Problem, we can choose at most one action from each  $\mathcal{B}_i$ . This can be handled in the same way. So again we can find a policy  $\sigma$  with profit at least  $\text{OPT} - O(\varepsilon) \cdot \text{MAX} = (1 - O(\varepsilon)) \text{OPT}$ , where  $\text{OPT}$  denotes the expected profit of the optimal policy and  $\text{MAX} = \max_{I \in \mathcal{V}} \text{DP}_1(I, \mathcal{A}) = \text{DP}_1(0, \mathcal{A}) = \text{OPT}$ .

## 7 Stochastic Target

In this section, we consider the stochastic target problem and prove Theorem 1.6. Define the item set  $\mathcal{A} = \{1, 2, \dots, n\}$ . Let the value  $I_t$  be the total profits of the items in the knapsack at time period  $t$ . Then we set  $T = m$  and  $I_1 = 0$ . When we insert an item  $i$  into the knapsack and observe its profit realization, say  $X_i$ , we define the system dynamic functions to be

$$I_{t+1} = f(I_t, i) = I_t + X_i, \quad g(I_t, i) = 0, \quad \text{and } h(I_{T+1}) = \begin{cases} 1 & \text{if } I_{T+1} \geq \mathbb{T}, \\ 0 & \text{otherwise;} \end{cases} \quad (7.1)$$

for  $t = 1, 2, \dots, T$ . Then Assumption 1 (2,3) is immediately satisfied. But Assumption 1 (1) is not satisfied for that the value space  $\mathcal{V}$  is not of constant size. Hence, we need discretization.

We use the same discretization technique as in [18] for the Expected Utility Maximization. The main idea is as follows. Without loss of generality, we set  $\mathbb{T} = 1$ . For an item  $b$ , we say  $X_b$  is a big realization if  $X_b > \varepsilon^4$  and small otherwise. For a big realization of  $X_b$ , we simply define the discretized version of  $\tilde{X}_b$  as  $\lfloor \frac{X_b}{\varepsilon^5} \rfloor \varepsilon^5$ . For a small realization of  $X_b$ , we define  $\tilde{X}_b = 0$  if  $X_b < d$  and  $\tilde{X}_b = \varepsilon^4$  if  $d \leq X_b \leq \varepsilon^4$ , where  $d$  is a threshold such that  $\Pr[X_b \geq d | X_b \leq \varepsilon^4] \varepsilon^4 = \mathbb{E}[X_b | X_b \leq \varepsilon^4]$ . For more details, please refer to [18].

Let  $\mathbb{P}(\sigma, \pi, 1)$  be the expected objective value of the policy  $\sigma$  for the instance  $(\pi, 1)$ , where  $\pi = \{\pi_i\}$  denotes the set of reward distributions and 1 denotes the target. Let  $\tilde{\pi}$  be the discretized version of  $\pi$ . Then, we have following lemmas.

**Lemma 7.1.** *For any policy  $\sigma$ , there exists a canonical policy  $\tilde{\sigma}$  such that*

$$\mathbb{P}(\tilde{\sigma}, \tilde{\pi}, (1 - 2\varepsilon)) \geq \mathbb{P}(\sigma, \pi, 1) - O(\varepsilon).$$

**Lemma 7.2.** *For any canonical policy  $\tilde{\sigma}$ ,*

$$\mathbb{P}(\tilde{\sigma}, \pi, (1 - 2\varepsilon)) \geq \mathbb{P}(\tilde{\sigma}, \tilde{\pi}, 1) - O(\varepsilon).$$

The proof of the lemma can be found in Appendix C.

*The Proof of Theorem 1.6.* Suppose  $\sigma^*$  is the optimal policy with expected value  $\text{OPT} = \mathbb{P}(\sigma^*, \pi, 1)$ . Given an instance  $\pi$ , we compute the discretized distribution  $\tilde{\pi}$ . By Lemma 7.1, there exists a policy  $\tilde{\sigma}^*$  such that

$$\mathbb{P}(\tilde{\sigma}^*, \tilde{\pi}, (1 - 2\varepsilon)) \geq \mathbb{P}(\sigma^*, \pi, 1) - O(\varepsilon) = \text{OPT} - O(\varepsilon).$$

Now, we present a stochastic dynamic program for the instance  $(\tilde{\pi}, 1 - 2\varepsilon)$ . Define the value set  $\mathcal{V} = \{0, \varepsilon^5, 2\varepsilon^5, \dots, 1\}$ , the item set  $\mathcal{A} = \{1, 2, \dots, n\}$ ,  $T = m$  and  $I_1 = 0$ . When we insert an item  $i$  into the knapsack and observe its profit realization, say  $X_i$ , we define the system dynamic functions to be

$$I_{t+1} = f(I_t, i) = \min\{1, I_t + X_i\}, \quad g(I_t, i) = 0, \quad \text{and} \quad h(I_{T+1}) = \begin{cases} 1 & \text{if } I_{T+1} \geq 1 - 2\varepsilon, \\ 0 & \text{otherwise;} \end{cases} \quad (7.2)$$

for  $I_t \in \mathcal{V}$  and  $t = 1, 2, \dots, T$ . Then Assumption 1 is immediately satisfied. By Theorem 1.1, we can find a policy  $\sigma$  with value  $\mathbb{P}(\sigma, \tilde{\pi}, 1 - 2\varepsilon)$  at least

$$\text{OPT}_d - O(\varepsilon) \cdot \text{MAX} \geq \mathbb{P}(\tilde{\sigma}^*, \tilde{\pi}, (1 - 2\varepsilon)) - O(\varepsilon) = \text{OPT} - O(\varepsilon),$$

where  $\text{OPT}_d$  denotes the expected value of the optimal policy for instance  $(\tilde{\pi}, 1 - 2\varepsilon)$  and  $\text{MAX} = \max_{I \in \mathcal{V}} \text{DP}_1(I, \mathcal{A}) = \text{DP}_1(1, \mathcal{A}) = 1$ . By Lemma 7.2, we have

$$\mathbb{P}(\sigma, \pi, (1 - 4\varepsilon)) \geq \mathbb{P}(\sigma, \tilde{\pi}, (1 - 2\varepsilon)) - O(\varepsilon) \geq \text{OPT} - O(\varepsilon),$$

which completes the proof. □

## 8 Stochastic Blackjack Knapsack

In this section, we consider the stochastic blackjack knapsack and prove Theorem 1.7. Define the item set  $\mathcal{A} = \{1, 2, \dots, n\}$ . Denote  $I_t = (I_{t,1}, I_{t,2})$  and let  $I_{t,1}, I_{t,2}$  be the total sizes and total profits of the items in the knapsack at the time period  $t$  respectively. We set  $T = n$  and  $I_1 = (0, 0)$ . When we insert an item  $i$  into the knapsack and observe its size realization, say  $s_i$ , we define the system dynamics function to be

$$I_{t+1} = f(I_t, i) = (I_{t,1} + s_i, I_{t,2} + p_i), \quad g(I_t, i) = 0, \quad \text{and} \quad h(I_{T+1}) = \begin{cases} I_{T+1,2} & \text{if } I_{T+1,1} \leq \mathbb{C}, \\ 0 & \text{otherwise;} \end{cases} \quad (8.1)$$

for  $t = 1, 2, \dots, T$ . Then Assumption 1 (2,3) is immediately satisfied. But Assumption 1 (1) is not satisfied for that the value space  $\mathcal{V}$  is not of constant size. Hence, we need discretization. Unlike the stochastic target problem, we need to discretize the sizes and profits simultaneously.

Consider a given adaptive policy  $\sigma$ . For each node  $v \in \mathcal{T}_\sigma$ , we have  $P(v) = \sum_{i \in \mathcal{R}(v)} p_i$  where  $\mathcal{R}(v)$  is the realization path from root to  $v$ . Define  $\mathcal{D} = \{v \in \text{LF} : W(v) \leq \mathbb{C}\}$  where LF is the set of leaves on  $\mathcal{T}_\sigma$ . Then we have

$$\mathbb{P}(\sigma) = \sum_{v \in \mathcal{D}} \Phi(v) \cdot P(v). \quad (8.2)$$

Without loss of generality, we assume  $\mathbb{C} = 1$  and  $X_i \in [0, 1]$  for any  $i \in [n]$ . Let  $\mathbb{P}(\sigma, \pi, 1)$  be the expected profit of the policy  $\sigma$  for the instance  $(\pi, 1)$ , where  $\pi = \{\pi_i\}$  denotes the set of size distributions and 1 denotes the capacity.

## 8.1 Discretization

Next, we show that item profits can be assumed to be bounded  $\theta_2 = \text{OPT}/\varepsilon^2$ . We set  $\theta_1 = \text{OPT}/\varepsilon$  and  $\theta_3 = \text{OPT}/\varepsilon^3$ . Now, we define an item to be a *huge profit* item if it has profit greater than or equal to  $\theta_2$ . We use the same discretization technique as in [6] for the stochastic knapsack. For a huge item  $b_i$  with size  $X_i$  and profit  $p_i$ , we define a new size  $\hat{X}_i$  and profit  $\hat{p}_i$  as follows: for  $\forall s \leq 1$

$$\Pr[\hat{X}_i = s] = \Pr[X_i = s] \cdot \frac{p_i}{\theta_2}, \quad \Pr[\hat{X}_i = 1 + 4\varepsilon] = 1 - \sum_{s \leq 1} \Pr[\hat{X}_i = s] \quad (8.3)$$

and  $\hat{p}_i = \theta_2$ . In Lemma 8.2, we show that this transformation can be performed with only an  $O(\varepsilon)$  loss in the optimal profit. Before to prove the lemma, we need following useful lemma.

**Lemma 8.1.** *For any policy  $\sigma$  on instance  $(\pi, \mathbb{C})$ , there exists a policy  $\sigma'$  such that  $\mathbb{P}(\sigma', \pi, \mathbb{C}) = (1 - O(\varepsilon))\mathbb{P}(\sigma, \pi, \mathbb{C})$  and in any realization path, the sum of profit of items except the last item that  $\sigma'$  inserts is less than  $\theta_1$ .*

*Proof.* We interrupt the process of the policy  $\sigma$  on a node  $v$  when the first time that  $P(v) \geq \theta_1$  to get a new policy  $\sigma'$ , i.e., we have a truncation on the node  $v$  and do not add items (include  $v$ ) any more in the new policy  $\sigma'$ . Let  $F$  be the set of the nodes on which we have truncation. Then we have  $\sum_{v \in F} \Phi(v) \leq \varepsilon$ . Thus, the total profit loss is equal to  $\sum_{v \in F} \Phi(v)\text{OPT} \leq \varepsilon\text{OPT}$ .  $\square$

W.l.o.g, we assume that all (optimal or near optimal) policies  $\sigma$  considered in this section satisfy the following property.

(P3) In any realization path, the sum of profit of items except the last item that  $\sigma$  inserts is less than  $\theta_1$ .

**Lemma 8.2.** *Let  $\pi$  be the distribution of size and profit for items and  $\hat{\pi}$  be the scaled version of  $\pi$  by Equation (8.3). Then, the following statement holds:*

1. For any policy  $\sigma$ , there exists a policy  $\hat{\sigma}$  such that

$$\mathbb{P}(\hat{\sigma}, \hat{\pi}, \mathbb{C}) = (1 - O(\varepsilon))\mathbb{P}(\sigma, \pi, \mathbb{C}).$$

2. For any policy  $\sigma$ ,

$$\mathbb{P}(\sigma, \pi, \mathbb{C}) = (1 - O(\varepsilon))\mathbb{P}(\sigma, \hat{\pi}, \mathbb{C}).$$

*Proof of Lemma 8.2.* For the first result, by Lemma 8.1, there exists a policy  $\hat{\sigma}$  such that  $\mathbb{P}(\hat{\sigma}, \pi, \mathbb{C}) = (1 - O(\varepsilon))\mathbb{P}(\sigma, \pi, \mathbb{C})$  and in any realization path, there are at most one huge profit item and always at the end of the policy. For huge profit item  $v$ , the expected profit contributed by the realization path from root to  $v$  to  $\mathbb{P}(\hat{\sigma}, \pi, \mathbb{C})$  is

$$\Phi(v) \cdot \Pr[X_v \leq \mathbb{C} - W(v)] \cdot (P(v) + p_v).$$

In  $\mathbb{P}(\hat{\sigma}, \hat{\pi}, \mathbb{C})$  with scaled distributions on huge profit items, the expected profit contributed by the realization path from the root to  $v$  is

$$\begin{aligned} & \Phi(v) \cdot \Pr[\hat{X}_v \leq \mathbb{C} - W(v)] \cdot (P(v) + \theta_2) \\ &= \Phi(v) \cdot \left( \Pr[X_v \leq \mathbb{C} - W(v)] \cdot \frac{p_v}{\theta_2} \right) \cdot (P(v) + \theta_2). \end{aligned}$$

Since  $v$  is a huge profit item, we have  $p_v \geq \theta_2$ , which implies  $\frac{p_v}{\theta_2} \cdot (P(v) + \theta_2) \geq P(v) + p_v$ . This completes the proof of the first part.

Now, we prove the second part. By Property (P3), for a huge item  $v$ , we have  $P(v) \leq \text{OPT}/\varepsilon$ . Then we have

$$\frac{p_v}{\theta_2} \cdot (P(v) + \theta_2) = p_v \cdot \left( 1 + \frac{P(v)}{\theta_2} \right) \leq p_v \cdot (1 + \varepsilon) \leq (1 + \varepsilon)(p_v + P(v)).$$

This completes the proof of the second part.  $\square$

In order to discretize the profit, we define the approximate profit  $\tilde{\mathbb{P}}(\sigma, \hat{\pi}) = \sum_{v \in \mathcal{D}} \Phi(v) \cdot \tilde{P}(v)$  where

$$\tilde{P}(v) = \theta_3 \cdot \left[ 1 - \prod_{i \in \mathcal{R}(v)} \left( 1 - \frac{p_i}{\theta_3} \right) \right] \quad (8.4)$$

Lemma 8.3 below can be used to bound the gap between the approximate profit and the original profit.

**Lemma 8.3.** *For any adaptive policy  $\sigma$  for the scaled distribution  $\hat{\pi}$ , we have*

$$\mathbb{P}(\sigma, \hat{\pi}, \mathbb{C}) \geq \tilde{\mathbb{P}}(\sigma, \hat{\pi}, \mathbb{C}) \geq (1 - O(\varepsilon))\mathbb{P}(\sigma, \hat{\pi}, \mathbb{C}).$$

*Proof.* Fix a node  $v$  on the tree  $\mathcal{T}_\sigma$ . For the left side, we have

$$\tilde{P}(v) = \theta_3 \cdot \left[ 1 - \prod_{i \in \mathcal{R}(v)} \left( 1 - \frac{p_i}{\theta_3} \right) \right] \leq \theta_3 \cdot \left[ 1 - \left( 1 - \sum_{i \in \mathcal{R}(v)} \frac{p_i}{\theta_3} \right) \right] = \sum_{i \in \mathcal{R}(v)} p_i = P(v).$$

For the right side, we have

$$\begin{aligned} \tilde{P}(v) &= \theta_3 - \theta_3 \cdot \left[ \prod_{i \in \mathcal{R}(v)} \left( 1 - \frac{p_i}{\theta_3} \right) \right] \\ &\geq \theta_3 - \theta_3 \cdot \left[ 1 - \sum_{i \in \mathcal{R}(v)} \frac{p_i}{\theta_3} + \left( \sum_{i \in \mathcal{R}(v)} \frac{p_i}{\theta_3} \right)^2 \right] \\ &= \left( \sum_{i \in \mathcal{R}(v)} p_i \right) \cdot \left[ 1 - \frac{\sum_{i \in \mathcal{R}(v)} p_i}{\theta_3} \right] \\ &\geq (1 - O(\varepsilon))P(v), \end{aligned}$$

where the last inequality holds by Property (P3) that  $P(v) \leq \theta_1 + \theta_2$ .  $\square$

Now, we choose the same discretization technique to discretize the sizes with parameter  $\varepsilon^3$  which is used in Section 7. For an item  $b$ , we say  $X_b$  is a big realization if  $X_b > \varepsilon^{4 \times 3}$  and small otherwise. For a big realization of  $X_b$ , we simple define the discretized version of  $\tilde{X}_b$  as  $\lfloor \frac{X_b}{\varepsilon^{5 \times 3}} \rfloor \varepsilon^{5 \times 3}$ . For a small realization of  $X_b$ , we define  $\tilde{X}_b = 0$  if  $X_b < d$  and  $\tilde{X}_b = \varepsilon^{4 \times 3}$  if  $d \leq X_b \leq \varepsilon^{4 \times 3}$ , where  $d$  is a threshold such that  $\Pr[X_b \geq d \mid X_b \leq \varepsilon^{4 \times 3}] \varepsilon^{4 \times 3} = \mathbb{E}[X_b \mid X_b \leq \varepsilon^{4 \times 3}]$ . For more details, please refer to [18].

**Lemma 8.4.** *Let  $\hat{\pi}$  be the distribution of size and profit for items and be  $\tilde{\pi}$  be the discretized version of  $\hat{\pi}$ . Then, the following statements holds:*

1. *For any policy  $\sigma$ , there exists a canonical policy  $\tilde{\sigma}$  such that*

$$\mathbb{P}(\tilde{\sigma}, \tilde{\pi}, (1 + 2\varepsilon)) \geq (1 - O(\varepsilon))\mathbb{P}(\sigma, \hat{\pi}, 1).$$

2. *For any canonical policy  $\tilde{\sigma}$ ,*

$$\mathbb{P}(\tilde{\sigma}, \hat{\pi}, (1 + 2\varepsilon)) \geq (1 - O(\varepsilon))\mathbb{P}(\tilde{\sigma}, \tilde{\pi}, 1).$$

## 8.2 Proof of Theorem 1.7

Now, we ready to prove Theorem 1.7.

*The proof of Theorem 1.7.* Suppose  $\sigma^*$  is the optimal policy with expected profit  $\text{OPT} = \mathbb{P}(\sigma^*, \pi, 1)$ . Given an instance  $\pi$ , we compute the scaled distribution  $\hat{\pi}$  and discretized distribution  $\tilde{\pi}$ . By Lemma 8.3, Lemma 8.4 (1) and Lemma 8.2 (1), there exist a policy  $\tilde{\sigma}^*$  such that

$$\begin{aligned} & \tilde{\mathbb{P}}(\tilde{\sigma}^*, \tilde{\pi}, (1 + 2\varepsilon)) \\ & \geq (1 - O(\varepsilon))\mathbb{P}(\tilde{\sigma}^*, \tilde{\pi}, (1 + 2\varepsilon)) \quad [\text{Lemma 8.3}] \\ & \geq (1 - O(\varepsilon))\mathbb{P}(\sigma^*, \hat{\pi}, 1) \quad [\text{Lemma 8.4 (1)}] \\ & \geq (1 - O(\varepsilon))\mathbb{P}(\sigma^*, \pi, 1) \quad [\text{Lemma 8.2 (1)}] \\ & = (1 - O(\varepsilon))\text{OPT}. \end{aligned}$$

Now, we present a stochastic dynamic program for the instance  $(\tilde{\pi}, 1 + 2\varepsilon)$ . Define the value set  $\mathcal{V} = \{0, \varepsilon^{5 \times 3}, 2\varepsilon^{5 \times 3}, \dots, 1 + 3\varepsilon\} \times \{0, 1\}$  and the item set  $\mathcal{A} = \{1, 2, \dots, n\}$ . We set  $T = n$  and  $I_1 = 0$ . When we insert an item  $i$  into the knapsack, we observe its size realization  $s_i$  and toss a coin to get a value  $\tilde{p}_i$  with  $\Pr[\tilde{p}_i = 1] = \frac{\hat{p}_i}{\theta_3}$  and  $\Pr[\tilde{p}_i = 0] = 1 - \frac{\hat{p}_i}{\theta_3}$ . Then we define the system dynamics function to be

$$I_{t+1} = f(I_t, i) = (I_{t+1,1}, I_{t+1,2}) = (\min\{1 + 3\varepsilon, I_{t,1} + s_i\}, \max\{I_{t,2}, \tilde{p}_i\}) \quad (8.5)$$

and  $g(I_t, i) = 0$  for  $I_t \in \mathcal{V}$  and  $t = 1, 2, \dots, T$ . The terminal function is

$$h(I_{T+1}) = \begin{cases} \theta_3 \cdot I_{T+1,2} & \text{if } I_{T+1} \leq 1 + 2\varepsilon, \\ 0 & \text{otherwise;} \end{cases} \quad (8.6)$$

Then Assumption 1 is immediately satisfied. By Theorem 1.1, we can find a policy  $\sigma$  with profit  $\tilde{\mathbb{P}}(\sigma, \tilde{\pi}, 1 + 2\varepsilon)$  at least

$$\text{OPT}_d - O(\varepsilon^4) \cdot \text{MAX} \geq \tilde{\mathbb{P}}(\tilde{\sigma}^*, \tilde{\pi}, (1 + 2\varepsilon)) - O(\varepsilon)\text{OPT} = (1 - O(\varepsilon))\text{OPT},$$

where  $\text{OPT}_d$  denotes the expected approximate profit of the optimal policy for instance  $(\tilde{\pi}, 1 + 2\varepsilon)$  and  $\text{MAX} = \max_{I \in \mathcal{V}} \text{DP}_1(I, \mathcal{A}) = \text{DP}_1((0, 1), \mathcal{A}) = \theta_3 = \frac{\text{OPT}}{\varepsilon^3}$ . By Lemma 8.3, Lemma 8.4 (2) and Lemma 8.2 (2), we have

$$\begin{aligned} & \mathbb{P}(\sigma, \pi, (1 + 4\varepsilon)) \\ & \geq (1 - O(\varepsilon))\mathbb{P}(\sigma, \hat{\pi}, (1 + 4\varepsilon)) \quad [\text{Lemma 8.2 (2)}] \\ & \geq (1 - O(\varepsilon))\mathbb{P}(\sigma, \tilde{\pi}, (1 + 2\varepsilon)) \quad [\text{Lemma 8.4 (2)}] \\ & \geq (1 - O(\varepsilon))\tilde{\mathbb{P}}(\sigma, \tilde{\pi}, (1 + 2\varepsilon)) \quad [\text{Lemma 8.3}] \\ & \geq (1 - O(\varepsilon))\text{OPT}, \end{aligned}$$

which completes the proof. □

### 8.3 Without Relaxing the Capacity

Before design a policy for SBK without relaxing the capacity  $\mathbb{C}$ , we establish a connection between adaptive policies for SKP and SBK. For a particular stochastic knapsack instance  $\mathcal{J}$ , we use  $\text{OPT}_{\text{SKP}}(\mathcal{J})$  to denote the expected profit of an optimal policy for stochastic knapsack. Similarly, we denote  $\text{OPT}_{\text{SBK}}(\mathcal{J})$  for stochastic blackjack knapsack. Note that a policy for SBK is also a policy for SKP.

**Lemma 8.5.** *For any policy  $\sigma$  for SKP on instance  $\mathcal{J} = (\pi, \mathbb{C})$ , there exists a policy  $\sigma'$  for SBK such that*

$$\mathbb{P}_{\text{SBK}}(\sigma', \pi, \mathbb{C}) \geq \frac{1}{4} \cdot \mathbb{P}_{\text{SKP}}(\sigma, \pi, \mathbb{C}). \quad (8.7)$$

*Proof.* W.l.o.g, we assume that for any node  $v \in \mathcal{T}_\sigma$ , we have  $\mathbb{P}(v) \leq \mathbb{P}_{\text{SKP}}(\sigma, \pi, \mathbb{C})$ . Otherwise, we use the subtree  $\mathcal{T}_v$  to instead  $\mathcal{T}_\sigma$  for SKP. Set  $\theta = \mathbb{P}_{\text{SKP}}(\sigma, \pi, \mathbb{C})/2$ . We interrupt the process of the policy  $\sigma$  on a node  $v$  when the first time that the summation of is larger than or equal to  $\theta$  to get a new policy  $\sigma'$ , *i.e.*, we have a truncation on the node  $v$  and do not insert the item (include  $v$ ) any more in the new policy  $\sigma'$ . Let  $F$  be the set of the nodes on which we have a truncation. Let  $\bar{F} = \text{LF} \setminus F$  be the set of rest leaves, where LF is the set of leaves of the tree  $\mathcal{T}_{\sigma'}$ . Then we have

$$\begin{aligned} 2\theta &= \sum_{v \in \bar{F}} \Phi(v) \cdot P(v) + \sum_{v \in F} \Phi(v) \cdot [P(v) + \mathbb{P}(v)] \\ &\leq \theta \cdot \sum_{v \in \bar{F}} \Phi(v) + \sum_{v \in F} \Phi(v)[P(v) + 2\theta] \\ &= \theta + \sum_{v \in F} \Phi(v)[P(v) + \theta] \\ &\leq \theta + 2 \sum_{v \in F} \Phi(v)P(v). \end{aligned}$$

Thus the expect profit of the policy  $\sigma'$  for SBK is equal to

$$\sum_{v \in F} \Phi(v)P(v) \geq \frac{1}{2} \cdot \theta = \frac{1}{4} \cdot \mathbb{P}_{\text{SKP}}(\sigma, \pi, \mathbb{C}).$$

□

**Lemma 8.6.** *For any stochastic knapsack instance  $\mathcal{J}$ , we have*

$$\text{OPT}_{\text{SKP}}(\mathcal{J}) \geq \text{OPT}_{\text{SBK}}(\mathcal{J}) \geq \frac{1}{4} \text{OPT}_{\text{SKP}}(\mathcal{J}). \quad (8.8)$$

For any fixed  $\varepsilon \geq 0$  and instance  $\mathcal{J}$ , by the result of [5], there is a polynomial time algorithm to compute a policy  $\sigma$  for SKP with expected profit  $(\frac{1}{2} - \varepsilon)\text{OPT}_{\text{SKP}}(\mathcal{J})$ . By Lemma 8.5, we can find a policy  $\sigma'$  for SBK expected profit at least

$$\frac{1}{4} \times (\frac{1}{2} - \varepsilon)\text{OPT}_{\text{SKP}}(\mathcal{J}) \geq (\frac{1}{8} - \varepsilon)\text{OPT}_{\text{SBK}}(\mathcal{J}).$$

This completes the proof of Theorem 1.8.

## 9 Concluding Remarks

In the paper, we formally define a model based on stochastic dynamic programs. This is a generic model. There are a number of stochastic optimization problems which fit in this model. We design a polynomial time approximation schemes for this model.

We also study two important stochastic optimization problems, Probemax problem and stochastic knapsack problem. Using the stochastic dynamic programs, we design a PTAS for Probemax problem,

which improves the best known approximation ratio  $1 - 1/e$ . To improve the approximation ratio for Probemax with a matroid constraint is still an open problem.

Next, we focus on the variants of the stochastic knapsack problem: stochastic blackjack knapsack and stochastic target problem. Using the stochastic dynamic programming and discretization technique, we design a PTAS for them if allowed to relax the capacity or target. To improve the ratio for the stochastic knapsack problem and variants without relaxing the capacity is still an open problem.

## Acknowledgements

We would like to thank Anupam Gupta for several helpful discussions during the various stages of the paper. Jian Li would like to thank the Simons Institute for the Theory of Computing, where part of this research was carried out. Hao Fu would like to thank Sahil Singla for useful discussions about Pandora's Box problem. Pan Xu would like to thank Aravind Srinivasan for his many useful comments.

## References

- [1] Marek Adamczyk, Maxim Sviridenko, and Justin Ward. Submodular stochastic probing on matroids. *Mathematics of Operations Research*, 41(3):1022–1038, 2016.
- [2] Arash Asadpour and Hamid Nazerzadeh. Maximizing stochastic monotone submodular functions. *Management Science*, 62(8):2374–2391, 2015.
- [3] Richard Bellman. Dynamic programming. In *Princeton University Press*, 1957.
- [4] Dimitri P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.
- [5] Anand Bhalgat. A  $(2 + \epsilon)$ -approximation algorithm for the stochastic knapsack problem. *Unpublished Manuscript*, 2011.
- [6] Anand Bhalgat, Ashish Goel, and Sanjeev Khanna. Improved approximation results for stochastic knapsack problems. *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1647–1665, 2011.
- [7] Kai Chen and Sheldon M. Ross. An adaptive stochastic knapsack problem. *European Journal of Operational Research*, 239(3):625 – 635, 2014.
- [8] Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. Combinatorial multi-armed bandit with general reward functions. *Advances in Neural Information Processing Systems*, 2016.
- [9] Brian C Dean, Michel X Goemans, and Jan Vondrák. Adaptivity and approximation for stochastic packing problems. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 395–404. Society for Industrial and Applied Mathematics, 2005.
- [10] Hao Fu, Jian Li, and Pan Xu. A ptas for a class of stochastic dynamic programs. In *45th International Colloquium on Automata, Languages, and Programming*, 2018.
- [11] Anupam Gupta and Viswanath Nagarajan. A stochastic probing problem with applications. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 205–216. Springer, 2013.
- [12] Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Algorithms and adaptivity gaps for stochastic probing. pages 1731–1747, 2016.

- [13] Nir Halman, Diego Klabjan, Chung Lun Li, James Orlin, and David Simchi-Levi. Fully polynomial time approximation schemes for stochastic dynamic programs. In *Nineteenth Acm-Siam Symposium on Discrete Algorithms*, pages 700–709, 2008.
- [14] Nir Halman, Diego Klabjan, Chung-Lun Li, James Orlin, and David Simchi-Levi. Fully polynomial time approximation schemes for stochastic dynamic programs. *SIAM Journal on Discrete Mathematics*, 28(4):1725–1796, 2014.
- [15] Nir Halman, Giacomo Nannicini, and James Orlin. A computationally efficient fptas for convex stochastic dynamic programs. *SIAM Journal on Optimization*, 25(1):317–350, 2015.
- [16] Taylan İlhan, Seyed MR Iravani, and Mark S Daskin. The adaptive knapsack problem with stochastic rewards. *Operations Research*, 59(1):242–248, 2011.
- [17] Asaf Levin and Aleksander Vainer. Adaptivity in the stochastic blackjack knapsack problem. *Theoretical Computer Science*, 516:121–126, 2014.
- [18] Jian Li and Wen Yuan. Stochastic combinatorial optimization via poisson approximation. *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 971–980, 2013.
- [19] Will Ma. Improvements and generalizations of stochastic knapsack and markovian bandits approximation algorithms. *Mathematics of Operations Research*, 2017.
- [20] Kamesh Munagala. Approximation algorithms for stochastic optimization. <https://simons.berkeley.edu/talks/kamesh-munagala-08-22-2016-1>, Simons Institute for the Theory of Computing, 2016.
- [21] Warren B Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, volume 842. John Wiley & Sons, 2011.
- [22] David B Shmoys and Chaitanya Swamy. An approximation scheme for stochastic linear programming and its application to stochastic integer programs. *Journal of the ACM (JACM)*, 53(6):978–1012, 2006.
- [23] Sahil Singla. The price of information in combinatorial optimization. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2523–2532. SIAM, 2018.
- [24] Jan Vondrák, Chandra Chekuri, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 783–792. ACM, 2011.
- [25] Martin L. Weitzman. Optimal search for the best alternative. *Econometrica*, 47(3):641–654, 1979.

## A Proof of Lemma 4.1

**Lemma 4.1.** *Let  $\pi = \{\pi_i\}$  be the set of distributions of random variables and  $\tilde{\pi}$  be the discretized version of  $\pi$ . Then, we have:*

1. *For any policy  $\sigma$ , there exists a (canonical) policy  $\tilde{\sigma}$  such that*

$$\mathbb{P}(\tilde{\sigma}, \tilde{\pi}) \geq (1 - O(\varepsilon))\mathbb{P}(\sigma, \pi) - O(\varepsilon)\text{OPT};$$

2. *For any canonical policy  $\tilde{\sigma}$ ,*

$$\mathbb{P}(\tilde{\sigma}, \pi) \geq \mathbb{P}(\tilde{\sigma}, \tilde{\pi}).$$



*Proof of Lemma 4.1.* Recall that for each node  $v$  on the decision tree  $\mathcal{T}_\sigma$ , the value  $I_v$  is the maximum among the realized value of the probed items right before probing the item  $a_v$ . For a path  $\mathcal{R}$ , we use  $W(\mathcal{R})$  to denote the value of the last node on the path. Let  $E$  be the set of all root-to-leaf paths in  $\mathcal{T}_\sigma$ . Then we have

$$\mathbb{P}(\sigma, \pi) = \sum_{r \in E} \Phi(r) \cdot W(r). \quad (\text{A.1})$$

For the first result of Lemma 4.1, we prove that there is a randomized canonical policy  $\sigma_r$  such that  $\mathbb{P}(\sigma_r, \tilde{\pi}) \geq (1 - O(\varepsilon))\mathbb{P}(\sigma, \pi) - O(\varepsilon)\text{OPT}$ . Thus such a deterministic policy  $\tilde{\sigma}$  exists. Let  $\theta = \frac{\widetilde{\text{OPT}}}{\varepsilon}$  be a threshold. We interrupt the process of the policy  $\sigma$  on a node  $v$  when the first time we probe an item whose weight exceeds this threshold to get a new policy  $\sigma'$  *i.e.*, we have a truncation on the node  $v$  and do not probe items (include  $v$ ) any more in the new policy  $\sigma'$ . The total profit loss is equal to

$$\sum_{v \in LF} \Phi(v) \cdot [\mathbb{P}(v) - I_v] \leq \sum_{v \in LF} \Phi(v) \cdot \text{OPT} = \text{OPT} \times \sum_{v \in LF} \Phi(v) \leq O(\varepsilon) \cdot \text{OPT},$$

where  $LF$  is the set of the nodes on which we have a truncation. The last inequality holds because  $\text{OPT} \geq \sum_{v \in LF} \Phi(v) \cdot \mathbb{P}(v) \geq \theta \cdot \sum_{v \in LF} \Phi(v)$ .

The randomized policy  $\sigma_r$  is derived from  $\sigma'$  as follows.  $\mathcal{T}(\sigma_r, \tilde{\pi})$  has the same tree structure as  $\mathcal{T}(\sigma', \pi)$ . If  $\sigma_r$  probes an item  $\tilde{X}$  and observes a discretized size  $d \in \mathcal{V}$ , it chooses a random branch in  $\mathcal{T}(\sigma_r, \tilde{\pi})$  among those sizes that are mapped to  $d$ , *i.e.*,  $\{w_e \mid D_X(w_e) = d\}$  according to the probability distribution

$$\Pr[\text{branch } e \text{ is chosen}] = \frac{\Pr[X = w_e]}{\sum_{s \mid D_X(s) = d} \Pr[X = s]}.$$

Then by Equation (4.3), if  $w_e < \theta$  we have

$$\tilde{p}_e = \Pr[\tilde{X} = d] \cdot \Pr[\text{branch } e \text{ is chosen}] = p_e \cdot \frac{1 - \Pr[\tilde{X} = \theta]}{\Pr[X < \theta]}$$

and

$$\tilde{w}_e = \left\lfloor \frac{w_e}{\varepsilon \widetilde{\text{OPT}}} \right\rfloor \cdot \varepsilon \widetilde{\text{OPT}} \geq w_e - \varepsilon \widetilde{\text{OPT}}.$$

**Fact.** For any node  $v$  in the tree  $\mathcal{T}(\sigma', \pi)$  such that  $I_v < \theta$ , we have

$$\tilde{\Phi}(v) \geq (1 - O(\varepsilon))\Phi(v). \quad (\text{A.2})$$

When we regard the path  $\mathcal{R}(v)$  as a policy, the expected profit of the path  $\mathcal{R}(v)$  can obtain is at least

$$\theta \cdot \left[ 1 - \prod_{i \in \mathcal{R}(v)} (1 - \Pr[\tilde{X}_i = \theta]) \right],$$

which is less than  $\text{OPT}$ , where  $\mathcal{R}(v)$  is the path from the root to the node  $v$ . Then we have  $\prod_{i \in \mathcal{R}(v)} (1 - \Pr[\tilde{X}_i = \theta]) \geq 1 - O(\varepsilon)$ , which implies that

$$\tilde{\Phi}(v) = \Phi(v) \cdot \prod_{i \in \mathcal{R}(v)} \frac{1 - \Pr[\tilde{X}_i = \theta]}{\Pr[X_i \leq \theta]} \geq (1 - O(\varepsilon))\Phi(v).$$

Now we bound the profit that we can obtain from  $\mathcal{T}(\sigma_r, \tilde{\pi})$ . Let  $E$  be the set of all root-to-leaf paths in  $\mathcal{T}(\sigma', \pi)$ . We split it into two parts  $E_1 = \{r \in E : W(r) < \theta\}$  and  $E_2 = \{r \in E : W(r) \geq \theta\}$ . For the first part, we have

$$\sum_{r \in E_1} \tilde{\Phi}(r) \cdot \tilde{W}(r) \geq \sum_{r \in E_1} \tilde{\Phi}(r) \cdot [W(r) - O(\varepsilon \widetilde{\text{OPT}})]$$

$$\geq (1 - O(\varepsilon)) \left[ \sum_{r \in E_1} \Phi(r) \cdot W(r) \right] - O(\varepsilon \widetilde{\text{OPT}}).$$

As mentioned before, for any path  $r \in E_2$ , we interrupt the process of the policy  $\sigma$  when the first time we probe an item whose weight exceeds this threshold  $\theta$ . We use  $\ell_r$  to denote the item for path  $r$ . By Equation (4.2), we have  $\Pr[\tilde{X} = \theta] \cdot \theta = \Pr[X \geq \theta] \cdot \mathbb{E}[X \mid X \geq \theta]$ . Then, we have

$$\begin{aligned} \sum_{r \in E_2} \tilde{\Phi}(r) \cdot \tilde{W}(r) &= \sum_{r \in E_2} \tilde{\Phi}(\ell_r) \cdot \Pr[\tilde{X}_{\ell_r} = \theta] \cdot \theta \\ &= \sum_{r \in E_2} \tilde{\Phi}(\ell_r) \cdot \Pr[X_{\ell_r} \geq \theta] \cdot \mathbb{E}[X_{\ell_r} \mid X_{\ell_r} \geq \theta] \\ &\geq \sum_{r \in E_2} (1 - O(\varepsilon)) \Phi(\ell_r) \cdot \Pr[X_{\ell_r} \geq \theta] \cdot \mathbb{E}[X_{\ell_r} \mid X_{\ell_r} \geq \theta] \\ &= (1 - O(\varepsilon)) \sum_{r \in E_2} \Phi(r) \cdot W(r). \end{aligned}$$

In summation, the expected profit  $\mathbb{P}(\sigma_r, \tilde{\pi})$  is equal to

$$\begin{aligned} \sum_{r \in E} \tilde{\Phi}(r) \cdot \tilde{W}(r) &\geq (1 - O(\varepsilon)) \sum_{r \in E} \Phi(r) \cdot W(r) - O(\varepsilon \widetilde{\text{OPT}}) \\ &= (1 - O(\varepsilon)) \mathbb{P}(\sigma', \pi) - O(\varepsilon) \text{OPT} \\ &= (1 - O(\varepsilon)) \mathbb{P}(\sigma, \pi) - O(\varepsilon) \text{OPT}. \end{aligned}$$

Next, we prove the second result of Lemma 4.1. Recall that a canonical policy makes decisions based on the discretized sizes. Then  $\mathcal{T}(\tilde{\sigma}, \pi)$  has the same tree structure as  $\mathcal{T}(\tilde{\sigma}, \tilde{\pi})$ , except that it obtain the true profit rather than the discretized profit. By Equation (4.3), for an edge  $e$  with a weight  $\tilde{w}_e < \theta$  on  $\mathcal{T}(\tilde{\sigma}, \tilde{\pi})$ , we have

$$\pi_e = \sum_{s \in \mathcal{S}: D_X(s) = \tilde{w}_e} \Pr[X = s] \geq \tilde{\pi}_e.$$

**Fact.** For any node  $v$  in the tree  $\mathcal{T}(\tilde{\sigma}, \tilde{\pi})$  with  $I_v < \theta$ , we have

$$\tilde{\Phi}(v) \leq \Phi(v). \tag{A.3}$$

Similarly, we split the root-to-leaf paths set  $E$  into two parts  $E_1 = \{r \in E : \max_{e \in r} \tilde{w}_e < \theta\}$  and  $E_2 = \{r \in E : \max_{e \in r} \tilde{w}_e = \theta\}$ . Then, we have

$$\begin{aligned} \mathbb{P}(\tilde{\sigma}, \tilde{\pi}) &= \sum_{r \in E_1} \tilde{\Phi}(r) \cdot \tilde{W}(r) + \sum_{r \in E_2} \tilde{\Phi}(r) \cdot \tilde{W}(r) \\ &= \sum_{r \in E_1} \tilde{\Phi}(r) \cdot \tilde{W}(r) + \sum_{r \in E_2} \tilde{\Phi}(\ell_r) \cdot \Pr[\tilde{X}_{\ell_r} = \theta] \cdot \theta \\ &\leq \sum_{r \in E_1} \Phi(r) \cdot W(r) + \sum_{r \in E_2} \Phi(\ell_r) \cdot \Pr[X_{\ell_r} \geq \theta] \cdot \mathbb{E}[X_{\ell_r} \mid X_{\ell_r} \geq \theta] \\ &= \sum_{r \in E_1} \Phi(r) \cdot W(r) + \sum_{r \in E_2} \Phi(r) \cdot W(r) \\ &= \mathbb{P}(\tilde{\sigma}, \pi) \end{aligned}$$

□

## B Proof of Lemma 4.3

Lemma 4.3. Let  $\pi = \{\pi_i\}$  be the set of distributions of random variables and  $\tilde{\pi}$  be the discretized version of  $\pi$ . Then, we have:

1. For any policy  $\sigma$ , there exists a canonical policy  $\tilde{\sigma}$  such that

$$\mathbb{P}(\tilde{\sigma}, \tilde{\pi}) \geq (1 - O(\varepsilon))\mathbb{P}(\sigma, \pi) - O(\varepsilon)\text{OPT};$$

2. For any canonical policy  $\tilde{\sigma}$ ,

$$\mathbb{P}(\tilde{\sigma}, \pi) \geq \mathbb{P}(\tilde{\sigma}, \tilde{\pi}) - O(\varepsilon)\text{OPT}.$$

*Proof.* This can be proved by an analogous argument as Lemma 4.1. For the first result, we design a randomized canonical policy  $\sigma_r$  as before. Here,  $W(r)$  is the summation of the top- $k$  weights on the path  $r$ . For a root-to-leaf path  $r$ , the profit we get is equal to

$$W(r) = \max_{C \subseteq r, |C| \leq k} \left[ \sum_{i \in C} X_i \right]. \quad (\text{B.1})$$

Now we bound the profit we can obtain from  $\mathcal{T}(\sigma_r, \tilde{\pi})$ . recall that  $E_1 = \{r \in E : \max_{e \in r} w_e < \theta\}$  and  $E_2 = \{r \in E : \max_{e \in r} w_e \geq \theta\}$  where  $E$  is the set of all root-to-leaf paths. Then for any  $r \in E_1$ , we have

$$\widetilde{W}(r) \leq W(r) - k \cdot \varepsilon \widetilde{\text{OPT}} = W(r) - O(\varepsilon \widetilde{\text{OPT}}).$$

For the first part, we have

$$\sum_{r \in E_1} \tilde{\Phi}(r) \cdot \widetilde{W}(r) \geq (1 - O(\varepsilon)) \sum_{r \in E_1} [\Phi(r) \cdot W(r)] - O(\varepsilon \widetilde{\text{OPT}}).$$

For the second part, we have

$$\begin{aligned} \sum_{r \in E_2} \tilde{\Phi}(r) \cdot \widetilde{W}(r) &= \sum_{r \in E_2} \tilde{\Phi}(\ell_r) \cdot \Pr[\tilde{X}_{\ell_r} = \theta] \cdot (\theta + \widetilde{W}'(\ell_r)) \\ &\geq \sum_{r \in E_2} \tilde{\Phi}(\ell_r) \cdot \Pr[X_{\ell_r} \geq \theta] \cdot \left( \mathbb{E}[X_{\ell_r} | X_{\ell_r} \geq \theta] + \widetilde{W}'(\ell_r) \right) \\ &\geq \sum_{r \in E_2} \tilde{\Phi}(\ell_r) \cdot \Pr[X_{\ell_r} \geq \theta] \cdot \left( \mathbb{E}[X_{\ell_r} | X_{\ell_r} \geq \theta] + W'(\ell_r) - O(\varepsilon \widetilde{\text{OPT}}) \right) \\ &\geq (1 - O(\varepsilon)) \sum_{r \in E_2} \Phi(r) \cdot W(r) - O(\varepsilon \widetilde{\text{OPT}}) \end{aligned}$$

where  $W'(r)$  is the summation of top  $k - 1$  weights on the path  $r$ . In summation, the expected profit  $\mathbb{P}(\sigma_r, \tilde{\pi})$  is equal to

$$\sum_{r \in E} \tilde{\Phi}(r) \cdot \widetilde{W}(r) \geq (1 - O(\varepsilon))\mathbb{P}(\sigma, \pi) - O(\varepsilon)\text{OPT}.$$

Now, we prove the second result. Similarly, we have

$$\sum_{r \in E_1} \tilde{\Phi}(r) \cdot \widetilde{W}(r) \leq \sum_{r \in E_1} \Phi(r) \cdot W(r)$$

and

$$\begin{aligned} \sum_{r \in E_2} \tilde{\Phi}(r) \cdot \widetilde{W}(r) &= \sum_{r \in E_2} \tilde{\Phi}(\ell_r) \cdot \Pr[\tilde{X}_{\ell_r} = \theta] \cdot (\theta + \widetilde{W}'(\ell_r)) \\ &\leq \sum_{r \in E_2} \Phi(\ell_r) \cdot \Pr[X_{\ell_r} \geq \theta] \cdot \mathbb{E}[X_{\ell_r} | X_{\ell_r} \geq \theta] + O(\varepsilon)\text{OPT} \\ &\leq \sum_{r \in E_2} \Phi(r) \cdot W(r) + O(\varepsilon)\text{OPT} \end{aligned}$$

where the first inequality holds since  $\Pr[\tilde{X} = \theta] \leq \varepsilon$ . Hence, the proof of the lemma is completed.  $\square$

## C Proof of Lemma 7.1 and Lemma 7.2

**Lemma 7.1.** For any policy  $\sigma$ , there exists a canonical policy  $\tilde{\sigma}$  such that

$$\mathbb{P}(\tilde{\sigma}, \tilde{\pi}, (1 - 2\varepsilon)) \geq \mathbb{P}(\sigma, \pi, 1) - O(\varepsilon).$$

**Lemma 7.2.** For any canonical policy  $\tilde{\sigma}$ ,

$$\mathbb{P}(\tilde{\sigma}, \pi, (1 - 2\varepsilon)) \geq \mathbb{P}(\tilde{\sigma}, \tilde{\pi}, 1) - O(\varepsilon).$$

Consider a given adaptive policy  $\sigma$  and for each  $v \in \mathcal{T}_\sigma$ , let  $W(v)$  and  $\widetilde{W}(v)$  be the sum of rewards on the path  $\mathcal{R}(v)$  before and after discretization respectively. Recall that  $\Phi(v)$  is the probability associated with the path  $\mathcal{R}(v)$ . In the proof of Lemma 4.2 of [18], it shows that for any given set  $F$  of nodes in  $\mathcal{T}_\sigma$  which contains at most one node from each root-leaf path, our discretization has the below property:

$$\sum_{v \in F: |W(v) - \widetilde{W}(v)| \geq 2\varepsilon} \Phi(v) = O(\varepsilon). \quad (\text{C.1})$$

*The Proof of Lemma 7.1.* Consider a randomized canonical policy  $\tilde{\sigma}$  which has the same structure as  $\sigma$ . If  $\sigma_r$  inserts an item  $\tilde{X}$  and observes a discretized size  $d \in \mathcal{V}$ , it chooses a random branch in  $\mathcal{T}(\sigma_r, \tilde{\pi})$  among those sizes that are mapped to  $d$ , i.e.,  $\{w_e \mid D_X(w_e) = d\}$  according to the probability distribution

$$\Pr[\text{branch } e \text{ is chosen}] = \frac{\Pr[X = w_e]}{\sum_{s \mid D_X(s) = d} \Pr[X = s]}.$$

Then, the probability of an edge on  $\mathcal{T}_{\sigma_r}$  is the same as that of the corresponding edge on  $\mathcal{T}_\sigma$ . The only different is two edges are labels with different weight  $w_e$  on  $\mathcal{T}_\sigma$  and  $\tilde{w}_e$  on  $\mathcal{T}_{\sigma_r}$ .

Notice that  $\mathbb{P}(\tilde{\sigma}, \tilde{\pi}, (1 - 2\varepsilon))$  is the sum of all paths  $\mathcal{R}(v)$  with  $\widetilde{W}(v) \geq 1 - 2\varepsilon$ . Define  $\mathcal{D} = \{v \in \text{LF} : W(v) \geq 1\}$  and  $\tilde{\mathcal{D}} = \{v \in \text{LF} : \widetilde{W}(v) \geq 1 - 2\varepsilon\}$ , where LF is the set of leaves on  $\mathcal{T}(\sigma, \pi)$ . Therefore we have

$$\mathbb{P}(\sigma, \pi, 1) = \sum_{v \in \mathcal{D}} \Phi(v), \quad \mathbb{P}(\tilde{\sigma}, \tilde{\pi}, (1 - 2\varepsilon)) = \sum_{v \in \tilde{\mathcal{D}}} \Phi(v).$$

Consider the set  $\Delta_1 = \mathcal{D} \setminus \tilde{\mathcal{D}}$ . For each  $v \in \Delta_1$ , we have  $W(v) \geq 1$  and  $\widetilde{W}(v) < 1 - 2\varepsilon$ . Thus we claim that  $|W(v) - \widetilde{W}(v)| > 2\varepsilon$ , implying that  $\Delta_1 \subseteq \Delta \doteq \{v \in \text{LF} : |W(v) - \widetilde{W}(v)| \geq 2\varepsilon\}$ . Thus we have

$$\mathbb{P}(\tilde{\sigma}, \tilde{\pi}, (1 - 2\varepsilon)) \geq \mathbb{P}(\sigma, \pi, 1) - \sum_{v \in \Delta_1} \Phi(v) \geq \mathbb{P}(\sigma, \pi, 1) - \sum_{v \in \Delta} \Phi(v) \geq \mathbb{P}(\sigma, \pi, 1) - O(\varepsilon). \quad \square$$

*The Proof of Lemma 7.2.* In our case, we focus on the decision tree  $\mathcal{T}(\tilde{\sigma}, \tilde{\pi}, 1)$  and assume all  $\tilde{w}_e$  take discretized value.  $\mathcal{T}(\tilde{\sigma}, \tilde{\pi}, 1)$  has the same tree structure as  $\mathcal{T}(\tilde{\sigma}, \pi, 1 - 2\varepsilon)$ .

Define  $\Delta_2 = \{v \in \text{LF}, W(v) < 1 - 2\varepsilon, \widetilde{W}(v) \geq 1\}$ , where LF is the set of leaves in  $\mathcal{T}_\sigma$ . Then we see  $\widetilde{W}(v) - W(v) > 2\varepsilon$ , implying  $\Delta_2 \subseteq \Delta = \{v \in \text{LF} : |W(v) - \widetilde{W}(v)| \geq 2\varepsilon\}$ . By the result of Equation (C.1), we see

$$\sum_{v \in \Delta_2} \Phi(v) \leq \sum_{v \in \Delta} \Phi(v) = O(\varepsilon).$$

Therefore we claim that

$$\mathbb{P}(\tilde{\sigma}, \pi, (1 - 2\varepsilon)) \geq \mathbb{P}(\tilde{\sigma}, \tilde{\pi}, 1) - \sum_{v \in \Delta_2} \Phi(v) \geq \mathbb{P}(\tilde{\sigma}, \tilde{\pi}, 1) - O(\varepsilon). \quad \square$$

## D Proof of Lemma 8.4

**Lemma 8.4.** *Let  $\hat{\pi}$  be the distribution of size and profit for items and be  $\tilde{\pi}$  be the discretized version of  $\hat{\pi}$ . Then, the following statements hold:*

1. *For any policy  $\sigma$ , there exists a canonical policy  $\tilde{\sigma}$  such that*

$$\mathbb{P}(\tilde{\sigma}, \tilde{\pi}, (1 + 2\varepsilon)) \geq (1 - O(\varepsilon))\mathbb{P}(\sigma, \hat{\pi}, 1).$$

2. *For any canonical policy  $\tilde{\sigma}$ ,*

$$\mathbb{P}(\tilde{\sigma}, \hat{\pi}, (1 + 2\varepsilon)) \geq (1 - O(\varepsilon))\mathbb{P}(\tilde{\sigma}, \tilde{\pi}, 1).$$

*The proof of Lemma 8.4.* For the first result, consider a randomized canonical policy  $\tilde{\sigma}$  which has the same structure as  $\sigma$ . If  $\sigma_r$  inserts an item  $\tilde{X}$  and observes a discretized size  $d \in \mathcal{V}$ , it chooses a random branch in  $\mathcal{T}(\sigma_r, \tilde{\pi})$  among those sizes that are mapped to  $d$ , i.e.,  $\{w_e \mid D_X(w_e) = d\}$  according to the probability distribution

$$\Pr[\text{branch } e \text{ is chosen}] = \frac{\Pr[X = w_e]}{\sum_{s \mid D_X(s)=d} \Pr[X = s]}.$$

Then, the probability of an edge on  $\mathcal{T}_{\sigma_r}$  is the same as that of the corresponding edge on  $\mathcal{T}_{\sigma}$ . The only different is two edges are labels with different weight  $w_e$  on  $\mathcal{T}_{\sigma}$  and  $\tilde{w}_e$  on  $\mathcal{T}_{\sigma_r}$ .

We have  $P(v) = \sum_{i \in \mathcal{R}(v)} p_i$  which is less than  $O(\text{OPT}/\varepsilon)$  by Lemma 8.2. Define  $\mathcal{D} = \{v \in \text{LF} : W(v) \leq 1\}$  and  $\tilde{\mathcal{D}} = \{v \in \text{LF} : \tilde{W}(v) \leq 1 + 2\varepsilon\}$ , where LF is the set of leaves on  $\mathcal{T}(\sigma, \pi)$ . Then we have

$$\mathbb{P}(\sigma, \hat{\pi}, 1) = \sum_{v \in \mathcal{D}} \Phi(v) \cdot P(v), \quad \mathbb{P}(\tilde{\sigma}, \tilde{\pi}, 1 + 2\varepsilon) = \sum_{v \in \tilde{\mathcal{D}}} \Phi(v) \cdot P(v).$$

Define  $\Delta = \{v \in \text{LF} : |W(v) - \tilde{W}(v)| \geq 2\varepsilon\}$ . Then we have  $\mathcal{D} \setminus \tilde{\mathcal{D}} \subseteq \Delta$ . By the result of Equation (C.1), we have

$$\sum_{v \in \mathcal{D} \setminus \tilde{\mathcal{D}}} \Phi(v) \leq \sum_{v \in \Delta} \Phi(v) = O(\varepsilon^3).$$

By Property (P1), for any node  $v$ , we have  $P(v) \leq \theta_1 + \theta_2$ . Then the gap  $\mathbb{P}(\sigma, \pi, 1) - \mathbb{P}(\tilde{\sigma}, \tilde{\pi}, (1 + 2\varepsilon))$  is less than

$$\sum_{v \in \mathcal{D} \setminus \tilde{\mathcal{D}}} \Phi(v) \cdot P(v) \leq \sum_{v \in \mathcal{D} \setminus \tilde{\mathcal{D}}} \Phi(v) \cdot \frac{2\text{OPT}}{\varepsilon^2} = O(\varepsilon)\text{OPT}.$$

This completes the proof of the first part.

Now, we prove the second part. Since a canonical policy makes decisions based on the discretized,  $\mathcal{T}(\tilde{\sigma}, \tilde{\pi}, 1)$  has the same tree structure as  $\mathcal{T}(\tilde{\sigma}, \hat{\pi}, 1 + 2\varepsilon)$ . Define  $\mathcal{D} = \{v \in \text{LF} : W(v) \leq 1 + 2\varepsilon\}$  and  $\tilde{\mathcal{D}} = \{v \in \text{LF} : \tilde{W}(v) \leq 1\}$ , where LF is the set of leaves on  $\mathcal{T}(\tilde{\sigma}, \tilde{\pi}, 1)$ . Then we have

$$\mathbb{P}(\tilde{\sigma}, \hat{\pi}, 1 + 2\varepsilon) = \sum_{v \in \mathcal{D}} \Phi(v) \cdot P(v), \quad \mathbb{P}(\tilde{\sigma}, \tilde{\pi}, 1) = \sum_{v \in \tilde{\mathcal{D}}} \Phi(v) \cdot P(v).$$

Then the gap  $\mathbb{P}(\tilde{\sigma}, \tilde{\pi}, 1) - \mathbb{P}(\tilde{\sigma}, \hat{\pi}, (1 + 2\varepsilon))$  is equal to

$$\sum_{v \in \tilde{\mathcal{D}} \setminus \mathcal{D}} \Phi(v) \cdot P(v) \leq \sum_{v \in \tilde{\mathcal{D}} \setminus \mathcal{D}} \Phi(v) \cdot \frac{2\text{OPT}}{\varepsilon^2} = O(\varepsilon)\text{OPT}.$$

□